

FIG. 1

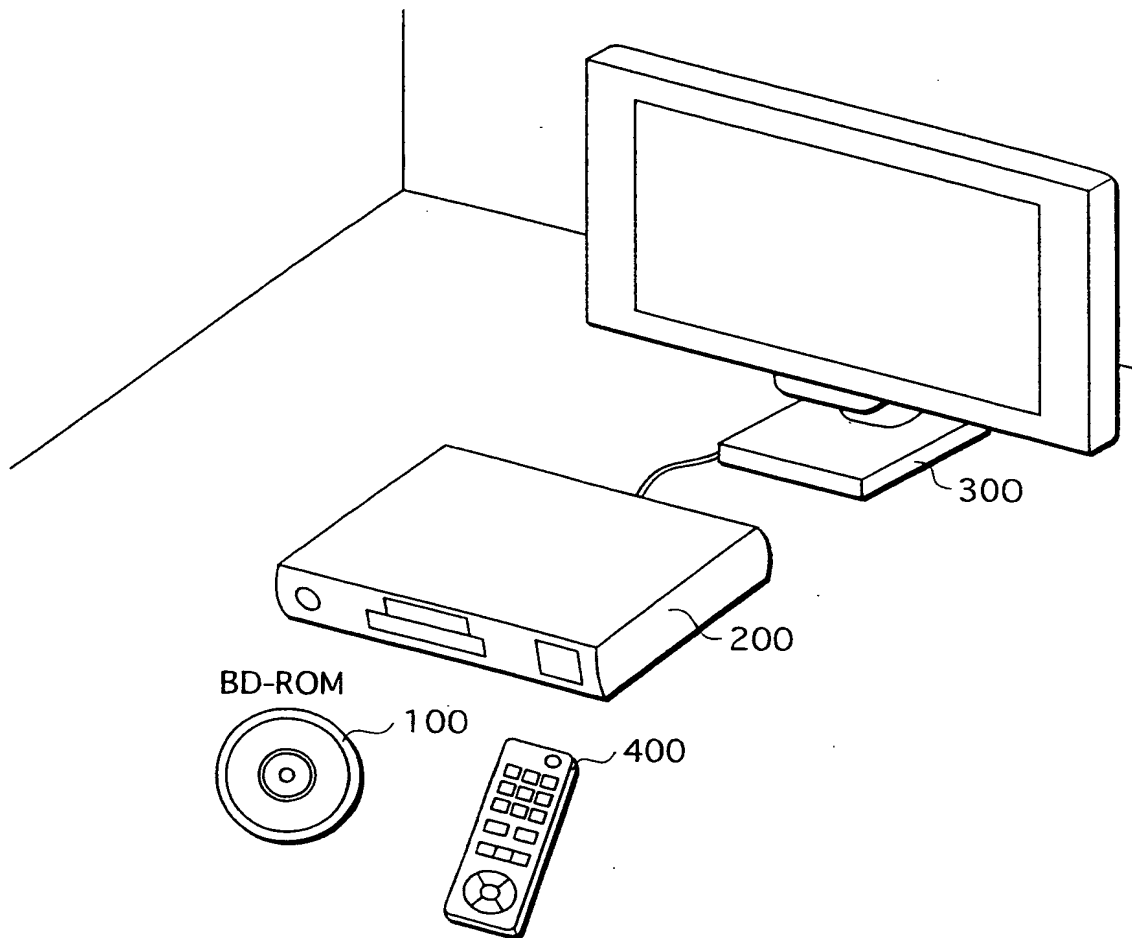


FIG.2

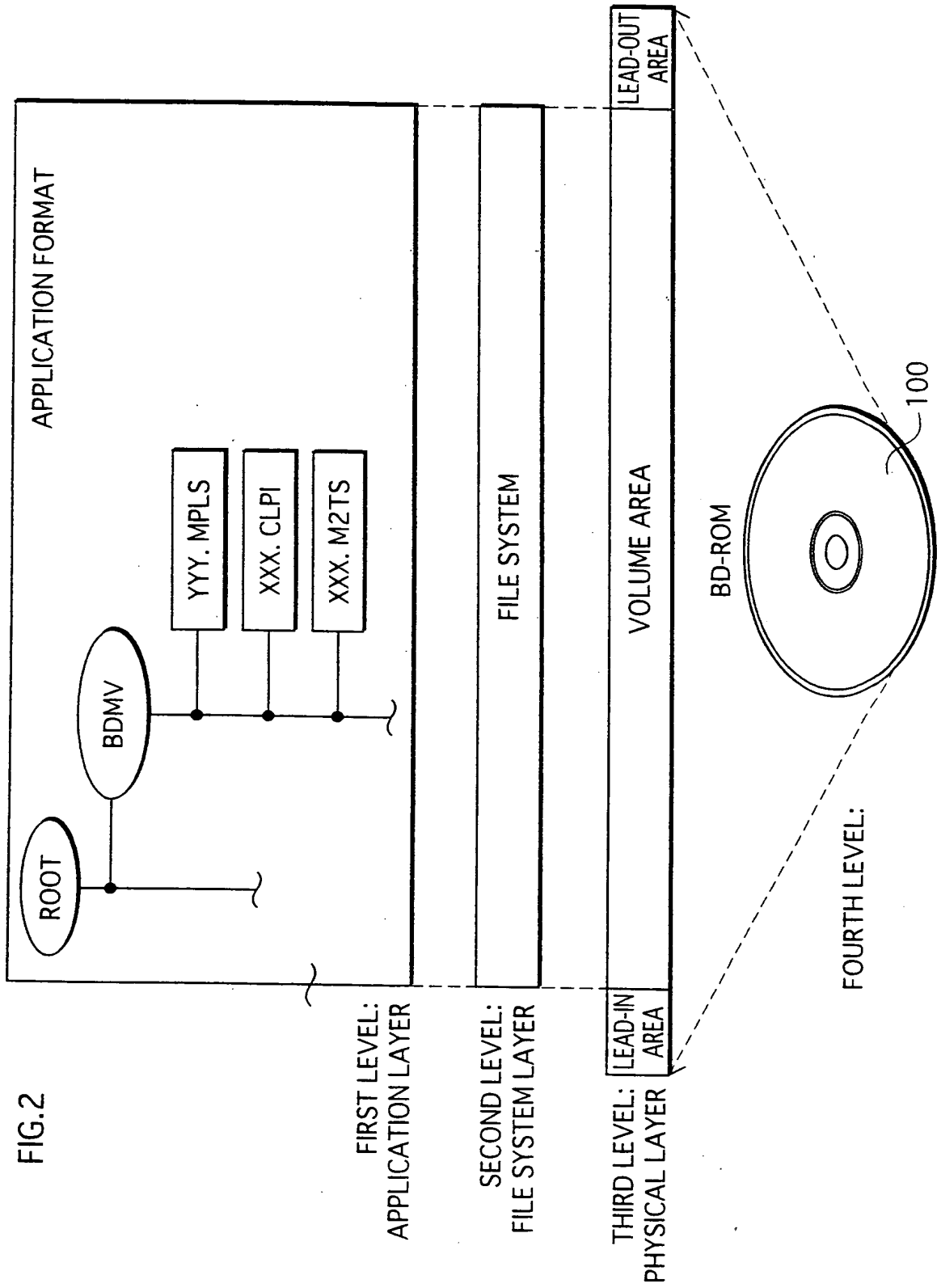


FIG. 3

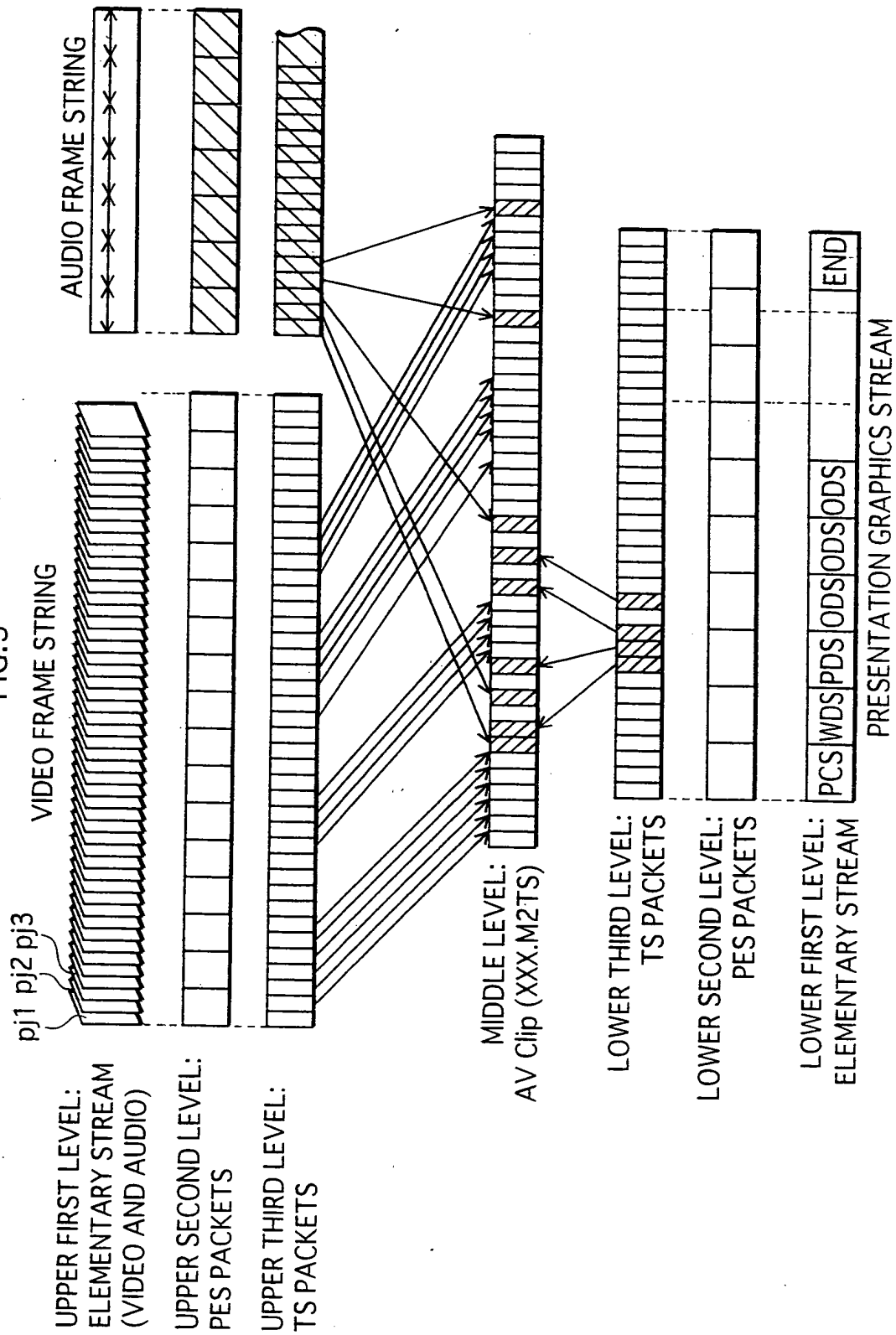


FIG. 4A

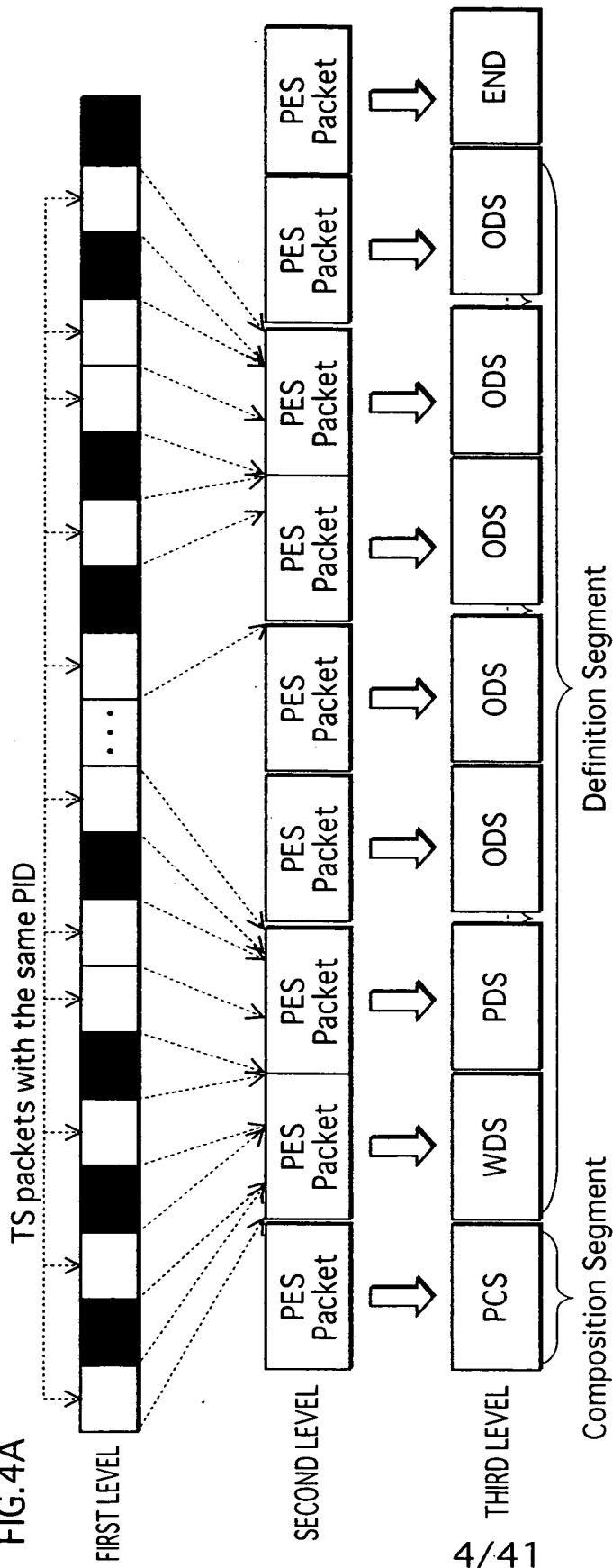


FIG. 4B

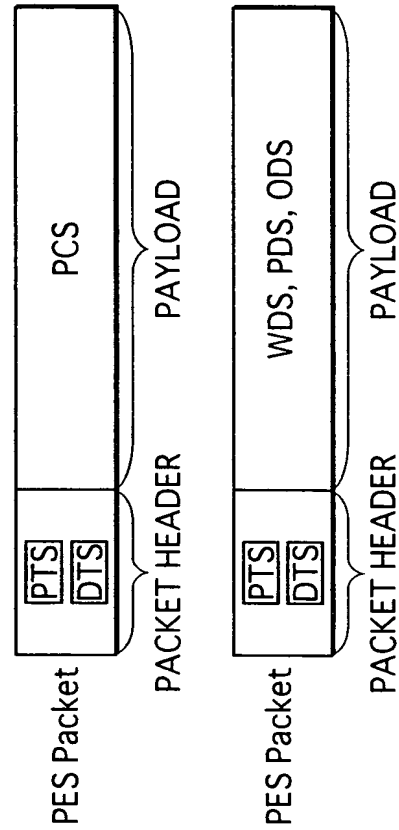


FIG. 5

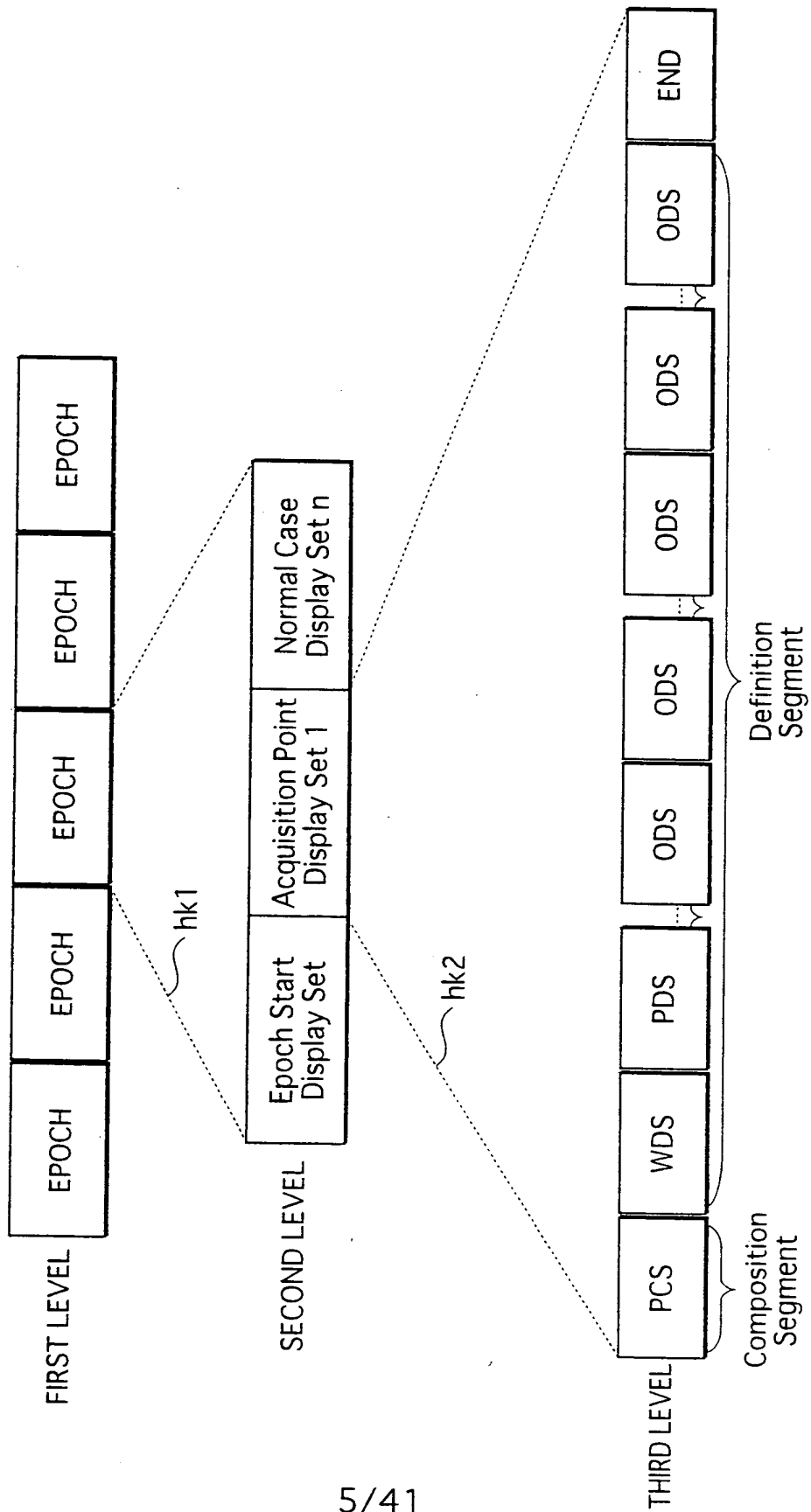


FIG.6

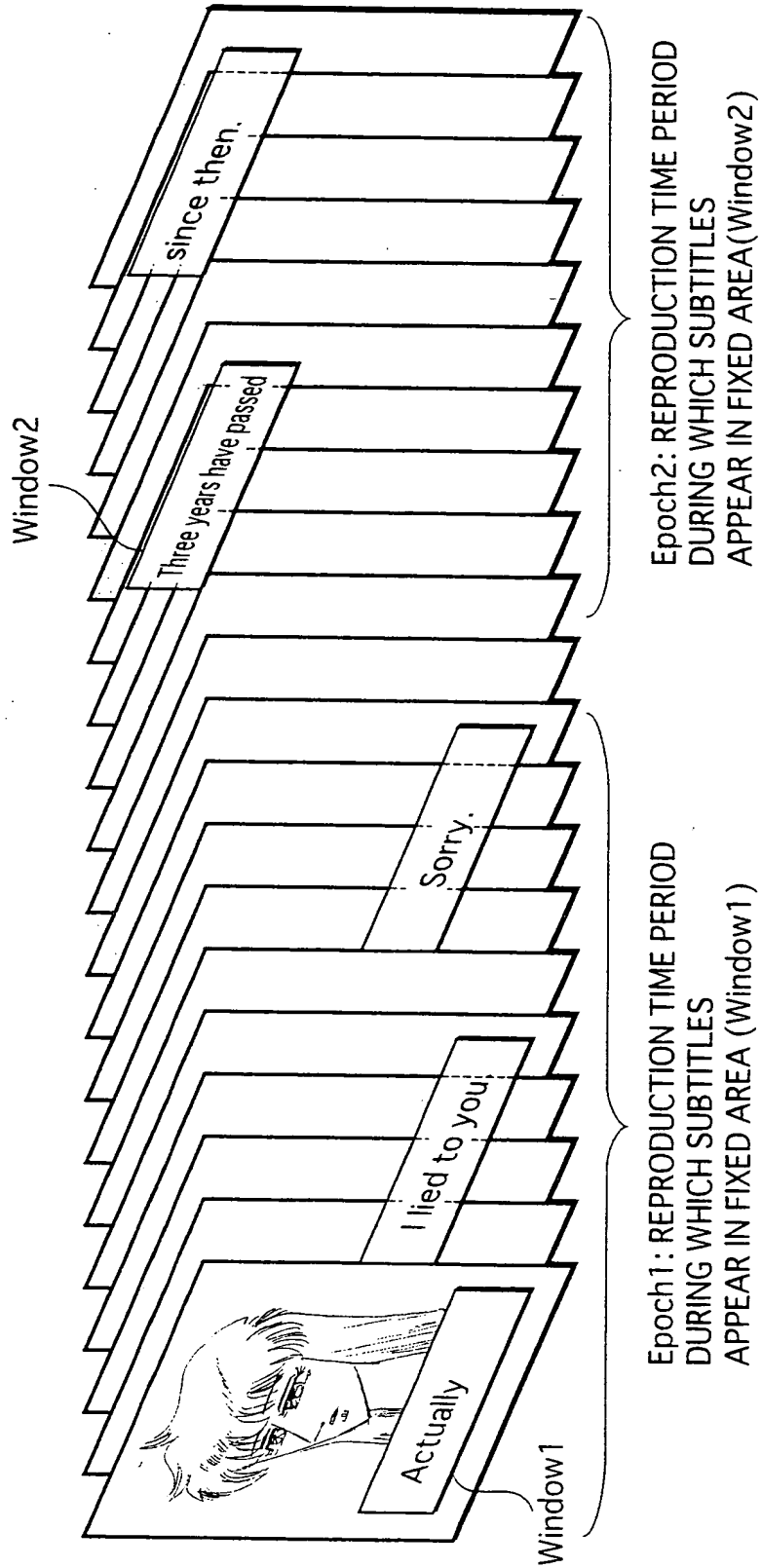


FIG.7A

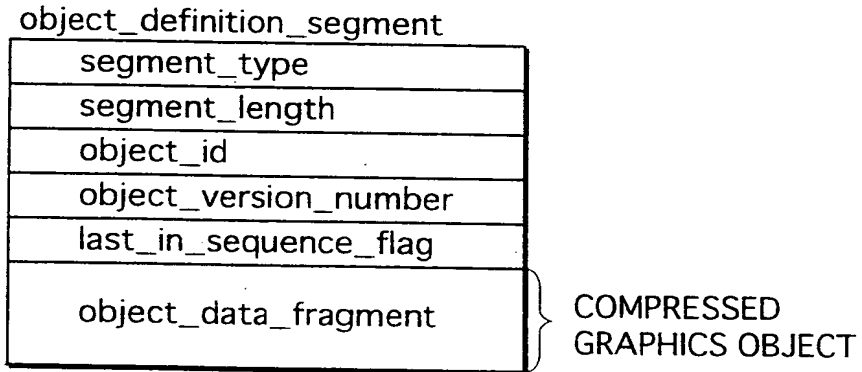


FIG.7B

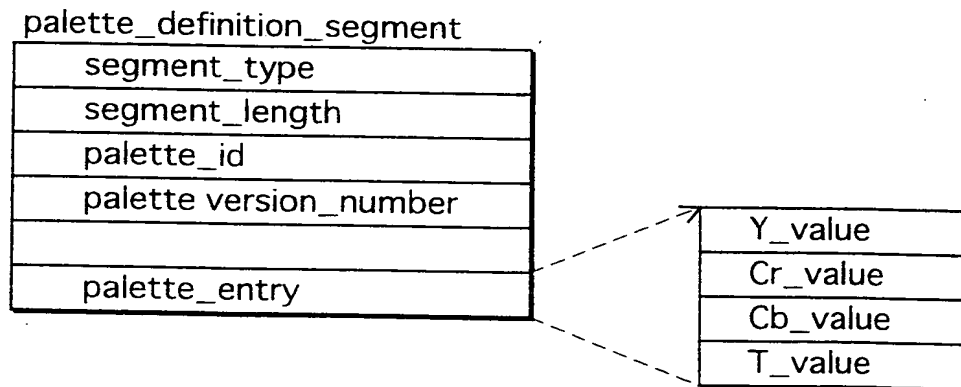


FIG.8A

window_definition_segment

window_id
window_horizontal_position
window_vertical_position
window_width
window_height

FIG.8B

presentation_composition_segment

segment_type
segment_length
composition_number
composition_state
palette_update_flag
palette_id_ref
composition_object(1)
composition_object(2)
:
composition_object(i)
:
composition_object(m)

wd1

object_id_ref
window_id_ref
object_cropped_flag
object_horizontal_position
object_vertical_position
cropping_rectangle_INFORMATION (1)
cropping_rectangle_INFORMATION (2)
:
cropping_rectangle_INFORMATION (i)
:
cropping_rectangle_INFORMATION (n)

wd2

object_cropping_horizontal_position
object_cropping_vertical_position
object_cropping_width
object_cropping_height

... REFERENCE VALUE IDENTIFYING
 GRAPHICS OBJECT READ IN
 ADVANCE

FIG. 9

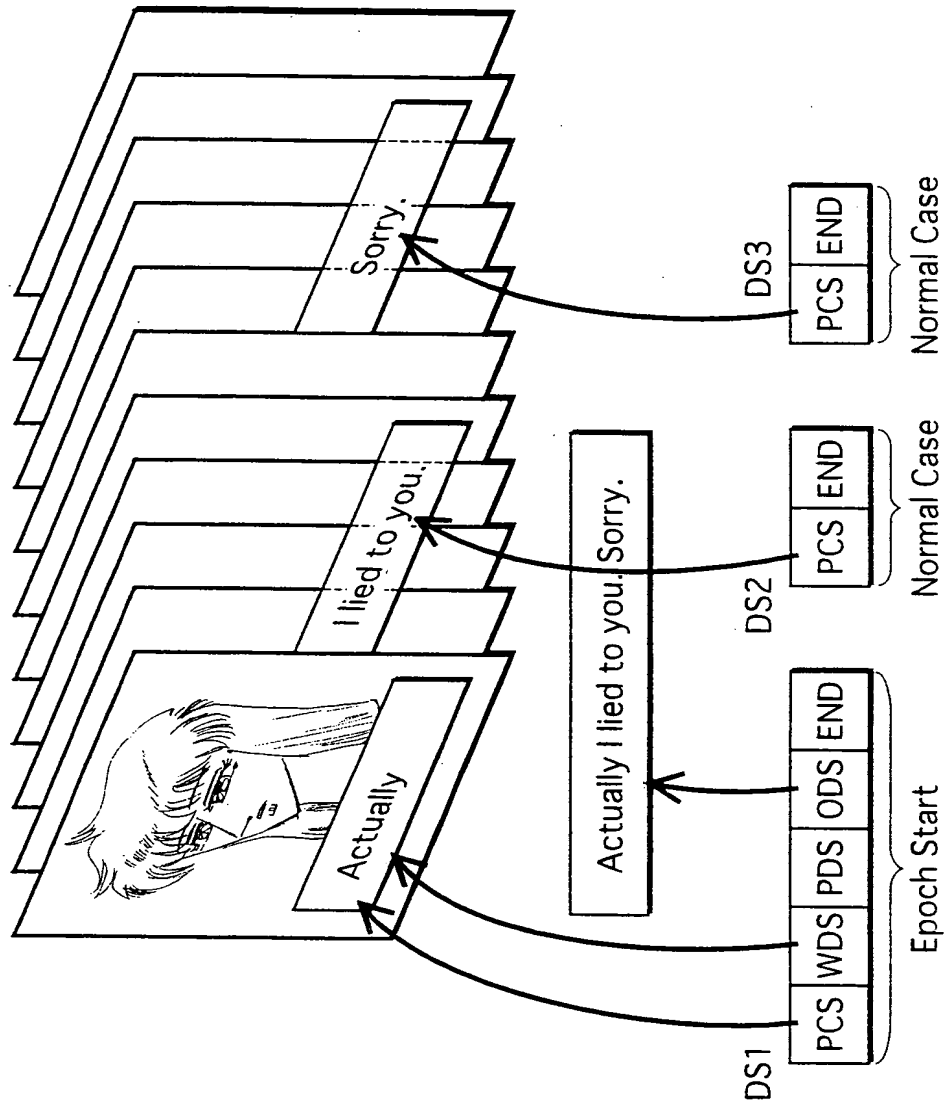


FIG.10

EXAMPLE DESCRIPTION
OF PCS AND WDS IN DS1

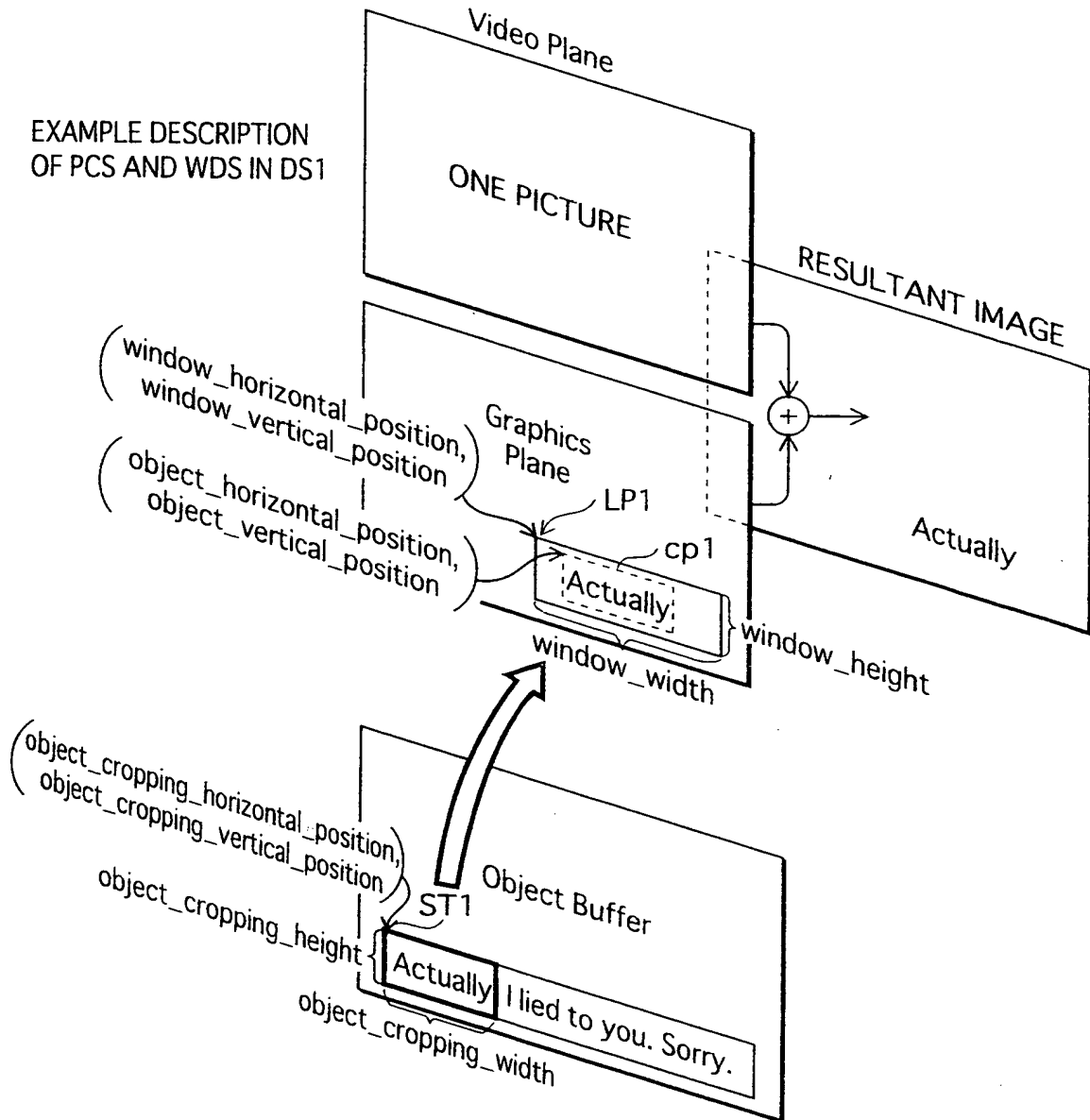


FIG. 11

EXAMPLE DESCRIPTION
OF PCS IN DS2

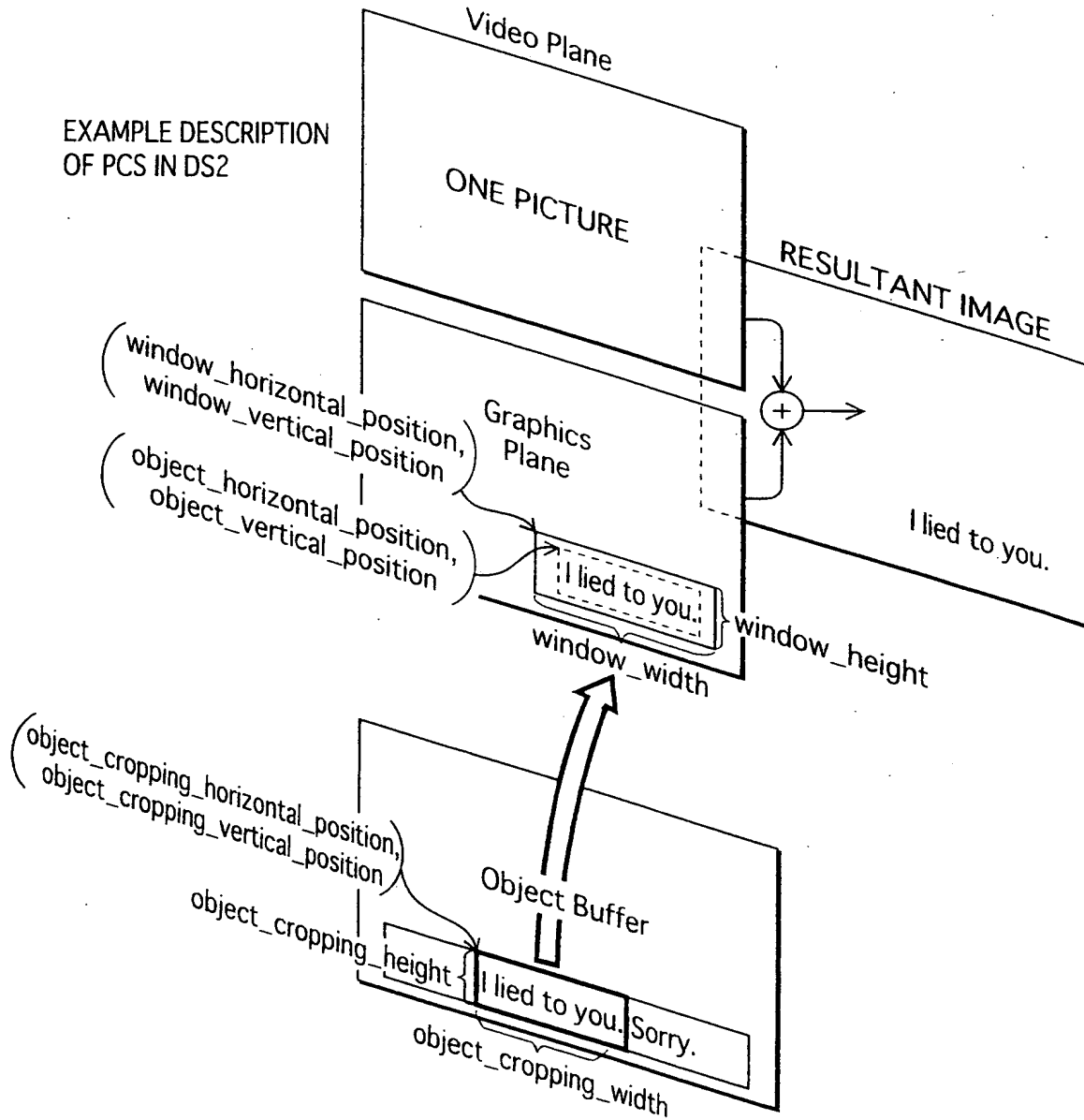


FIG. 12

EXAMPLE DESCRIPTION
OF PCS IN DS3

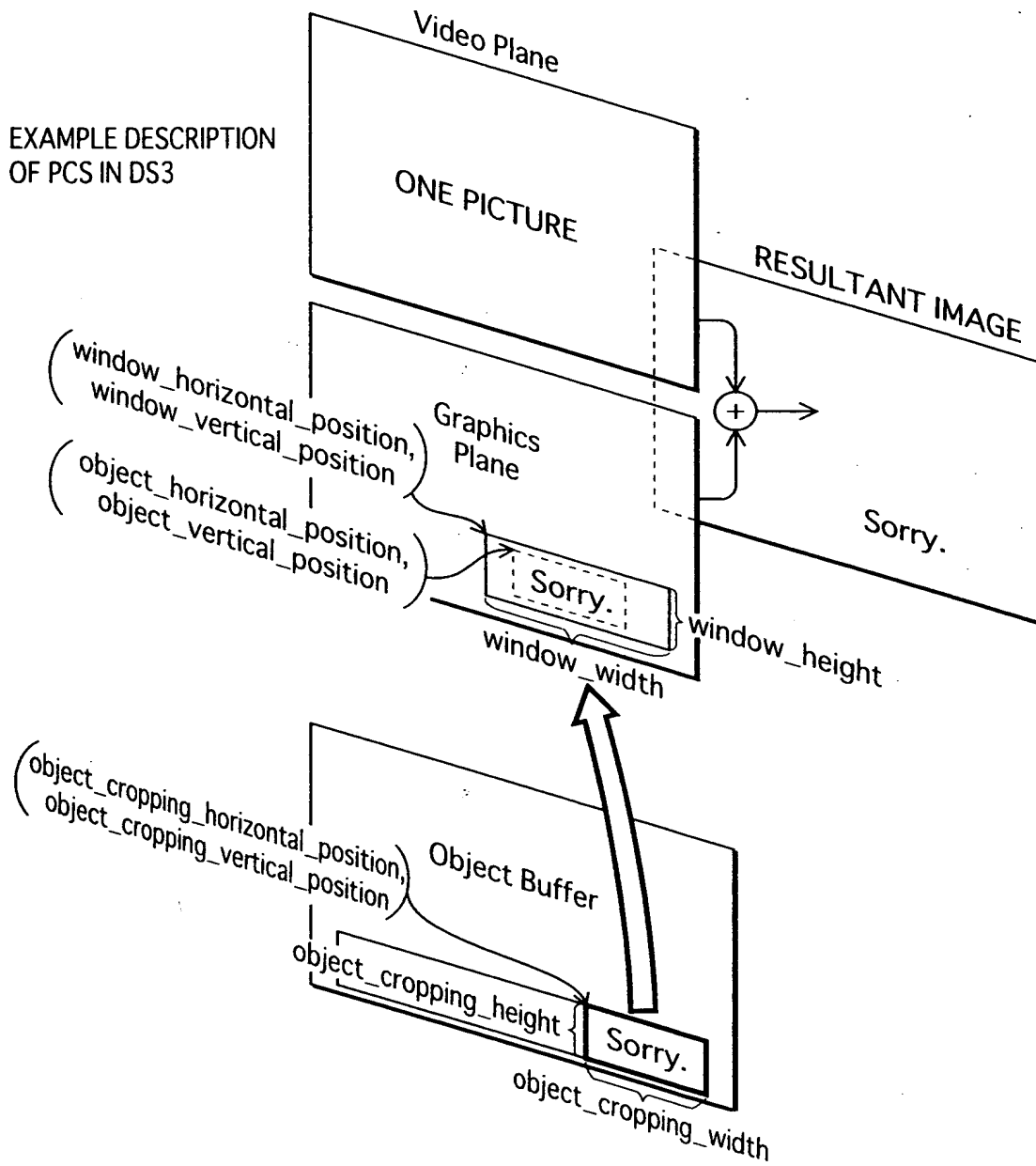


FIG. 13

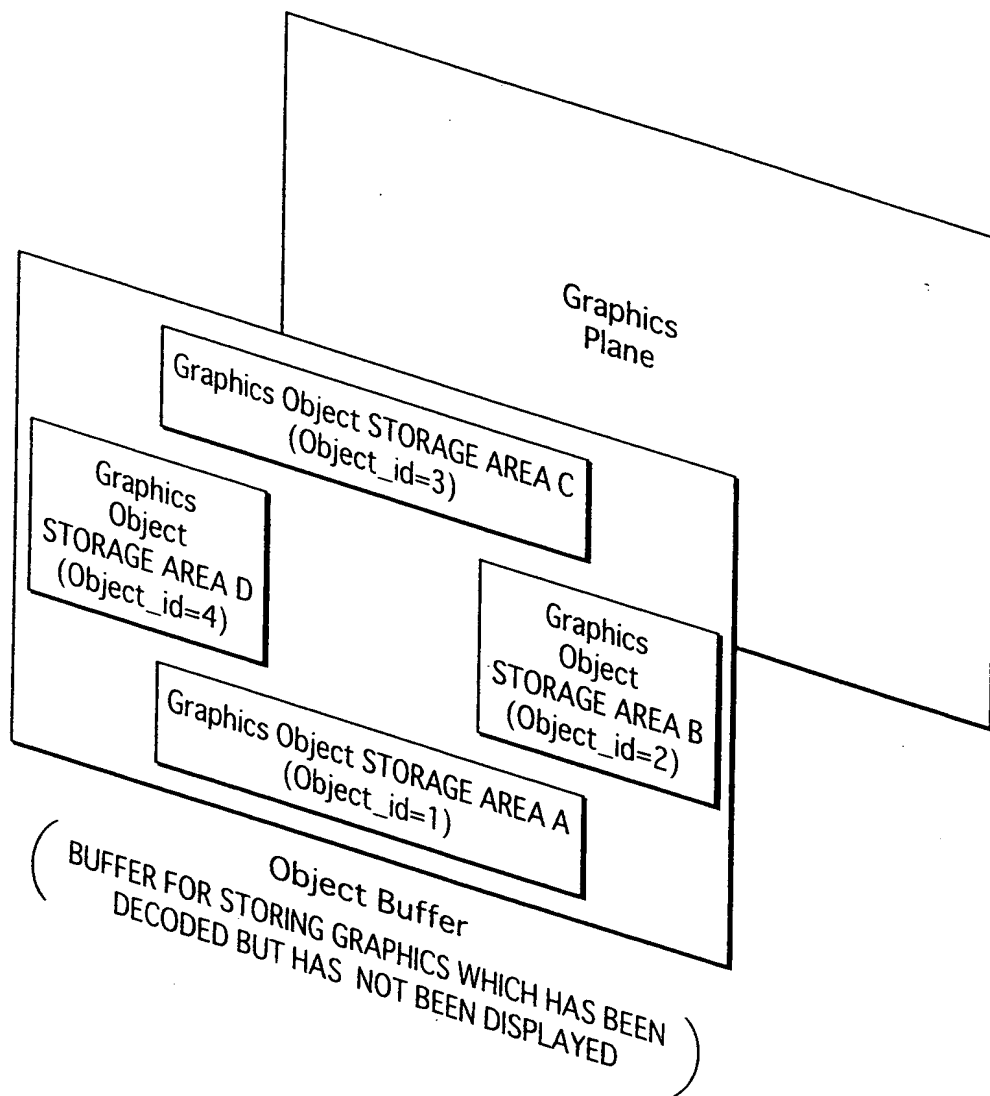


FIG. 14 $PTS(DS_n[PCS]) \geq DTS(DS_n[PCS]) + DECODEDURATION(DS_n)$

Where:

- $DECODEDURATION(DS_n)$ is calculated as follows:

```

decode_duration = 0 ;
decode_duration += PLANEINITIALIZATIONTIME( DS_n ) ;
if( DS_n.PCS.num_of_objects == 2 )
{
  decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[0], decode_duration ) ;
  if( DS_n.PCS.OBJ[0].window_id == DS_n.PCS.OBJ[1].window_id )
  {
    decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[1], decode_duration ) ;
    decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
  }
  else
  {
    decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
    decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[1], decode_duration ) ;
    decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[1].window_id )//256*106 ) ;
  }
}
else if( DS_n.PCS.num_of_objects == 1 )
{
  decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[0], decode_duration ) ;
  decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
}
return decode_duration ;

```

- $PLANEINITIALIZATIONTIME(DS_n)$ is calculated as follows:

```

initialize_duration=0 ;
if( DS_n.PCS.composition_state== EPOCH_START )
{
  initialize_duration = 90000*( 8*video_width*video_height//256*106 ) ;
}
else
{
  for( i=0 ; i < WDS.num_windows ; i++ )
  {
    if( EMPTY(DS_n.WDS.WIN[i],DS_n) )
      initialize_duration += 90000*( SIZE( DS_n.WDS.WIN[i] )//256*106 ) ;
  }
}
return initialize_duration ;

```

- $WAIT(DS_n, OBJ, current_duration)$ is calculated as follows:

```

wait_duration = 0 ;
if( EXISTS( OBJ.object_id, DS_n ) )
{
  object_definition_ready_time = PTS( GET( OBJ.object_id, DS_n ) ) ;
  current_time = DTS( DS_n.PCS )+current_duration ;
  if( current_time < object_definition_ready_time )
    wait_duration += object_definition_ready_time - current_time ;
}
return wait_duration ;

```

FIG. 15

CALCULATION OF DECODEDURATION

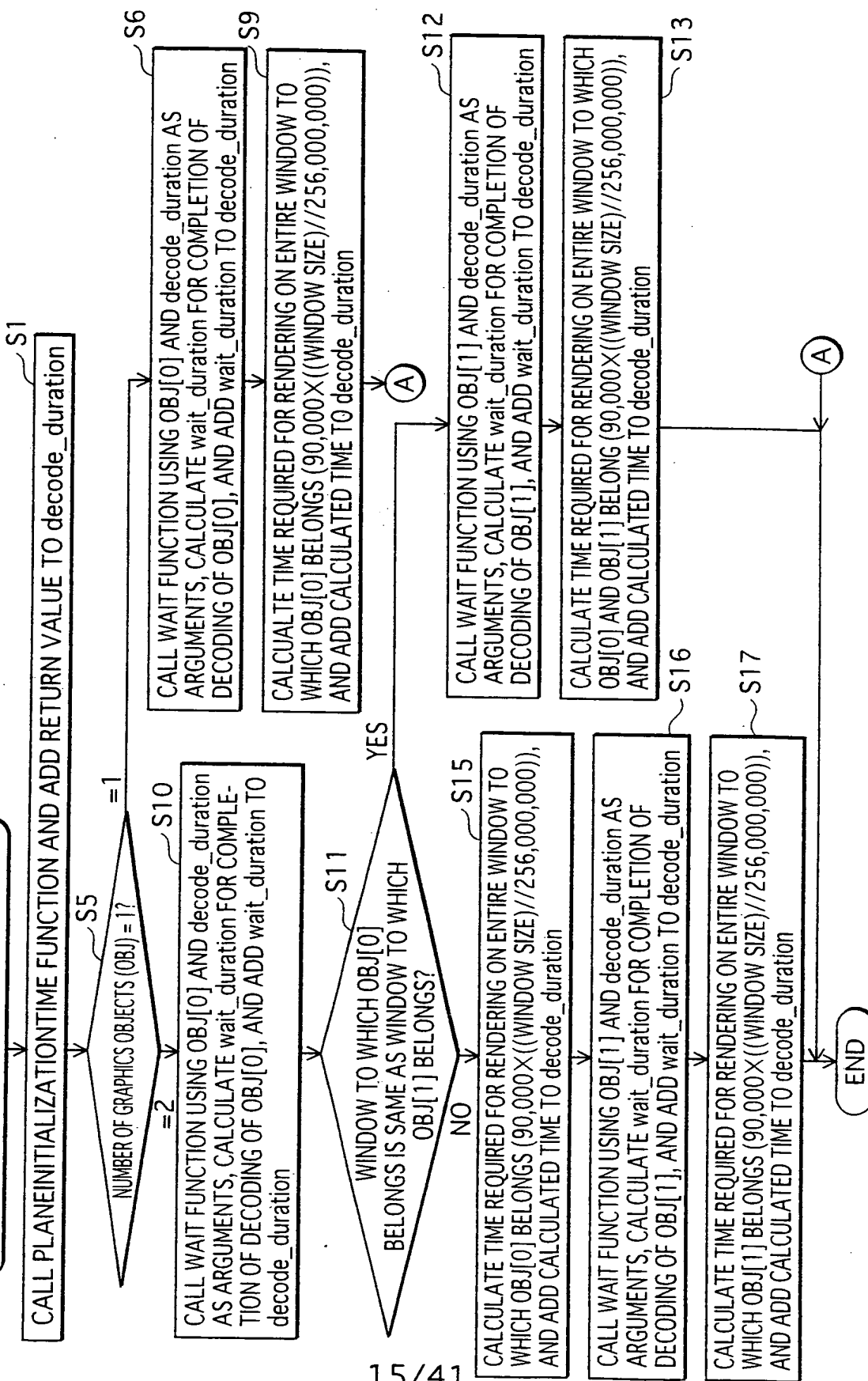


FIG.16A

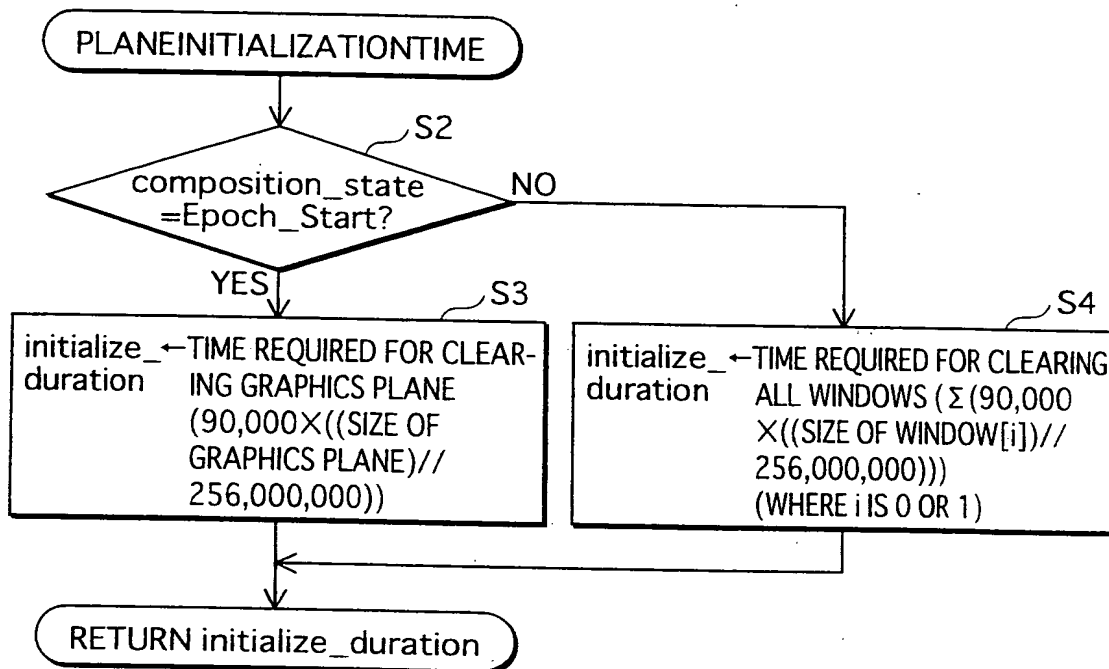


FIG.16B

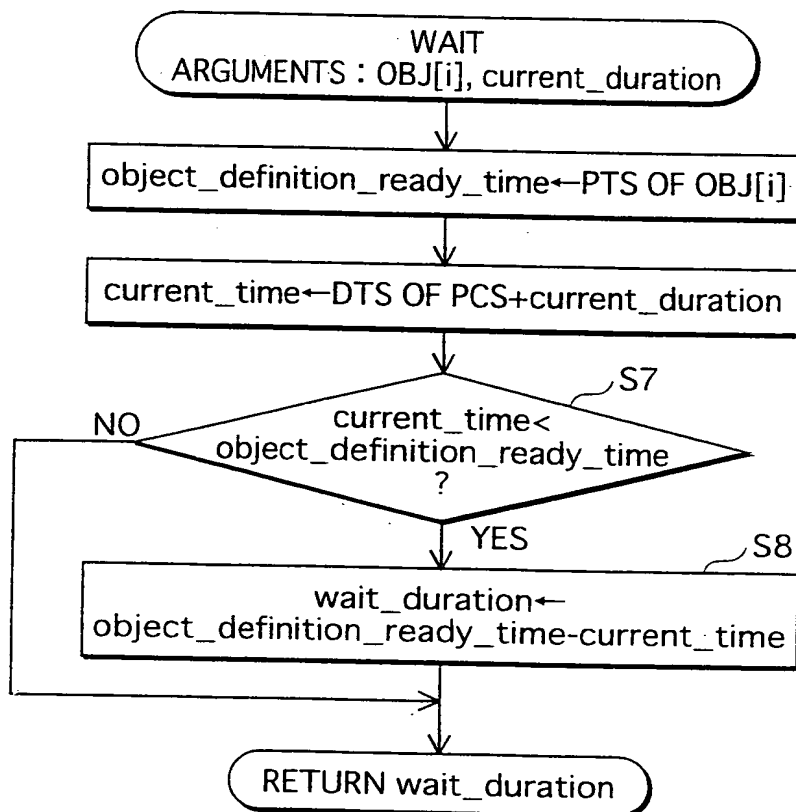


FIG.17A

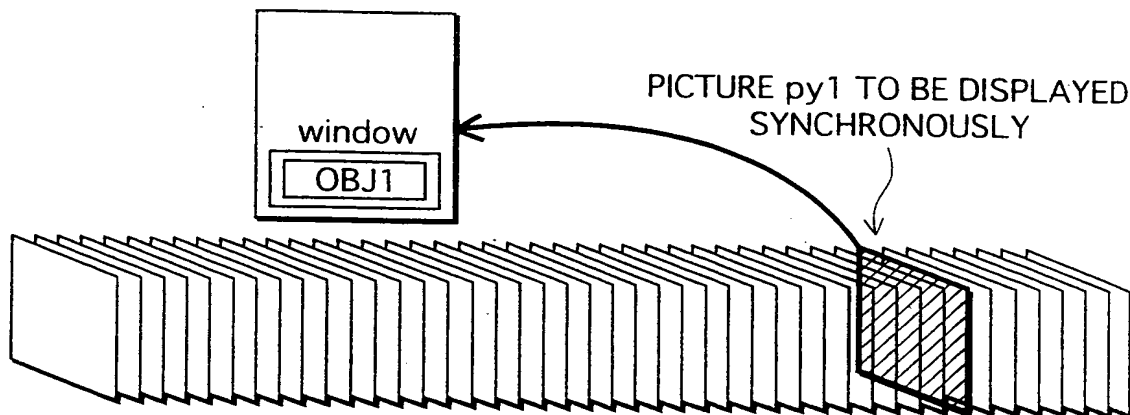


FIG.17B

DECODE_DURATION
 =(2)+(3)

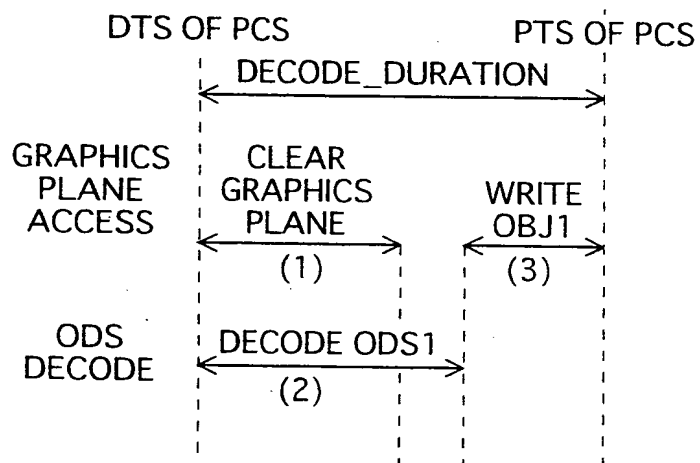


FIG.17C

DECODE_DURATION
 =(1)+(3)

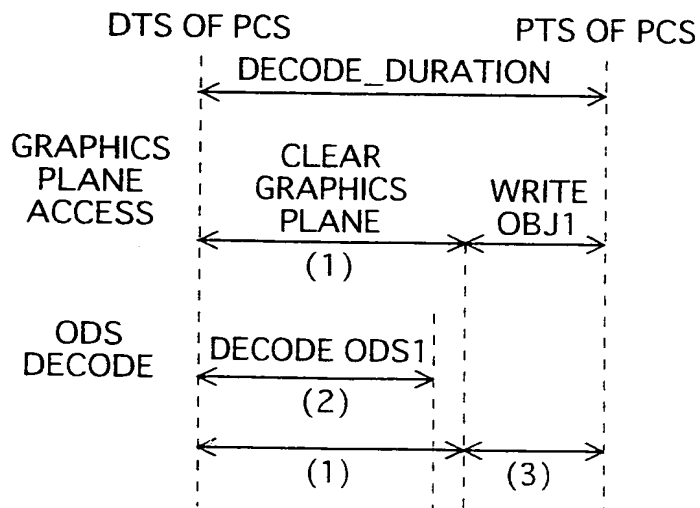


FIG.18A

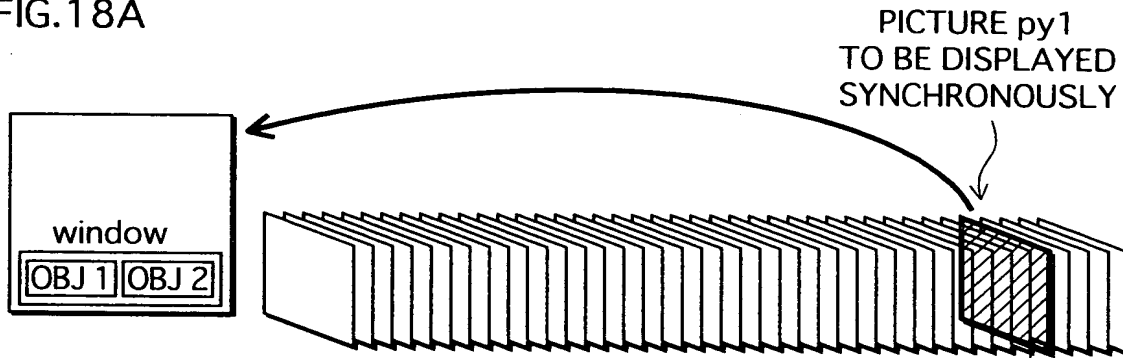


FIG.18B

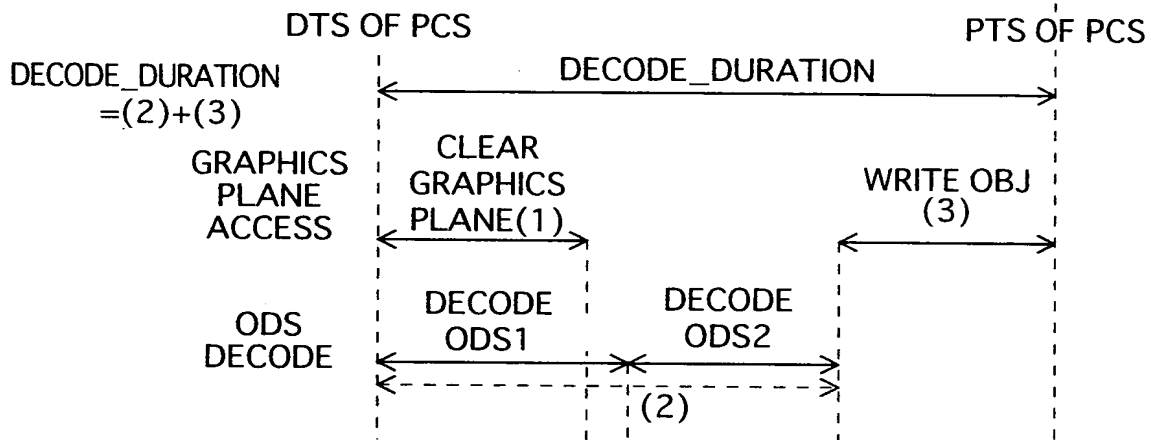
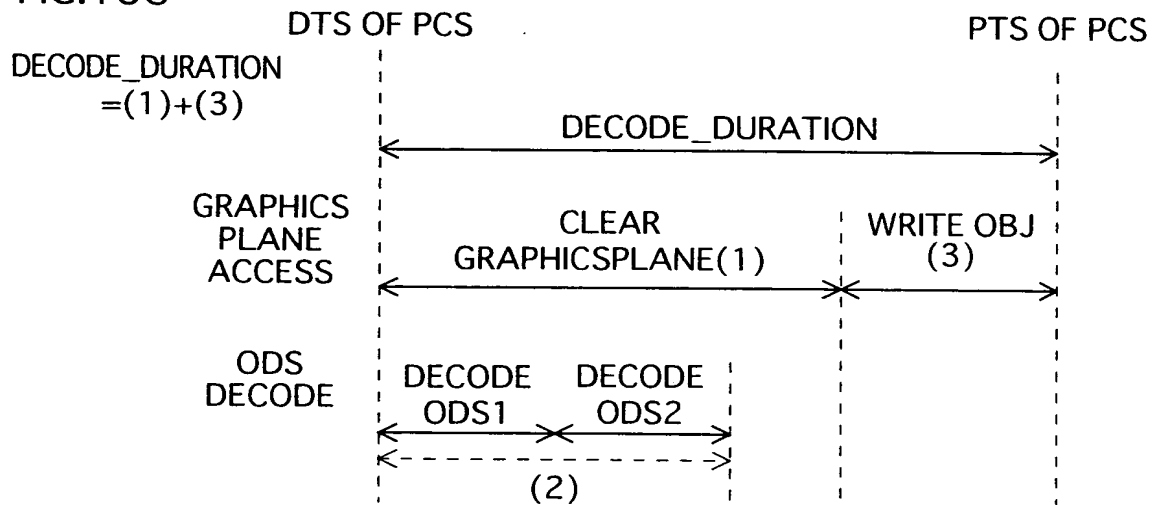


FIG.18C



PICTURE py1 TO BE DISPLAYED
 SYNCHRONOUSLY

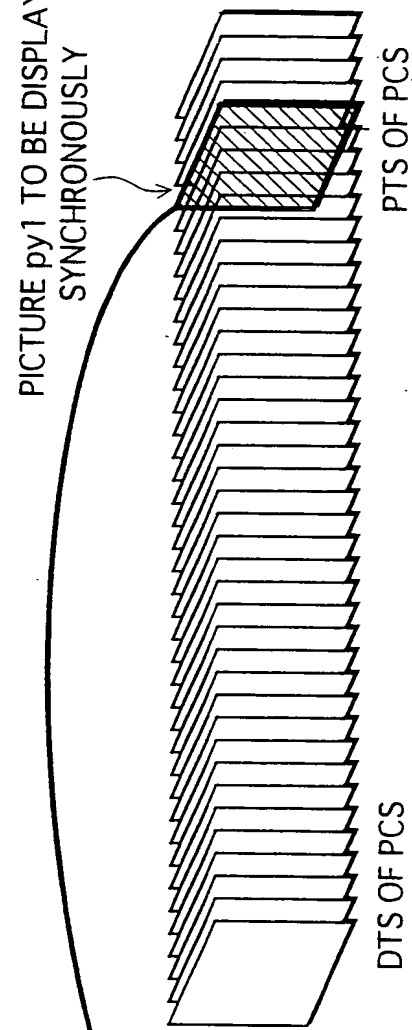


FIG.19A

FIG.19B

DECODE_DURATION
 $= (2) + (32)$

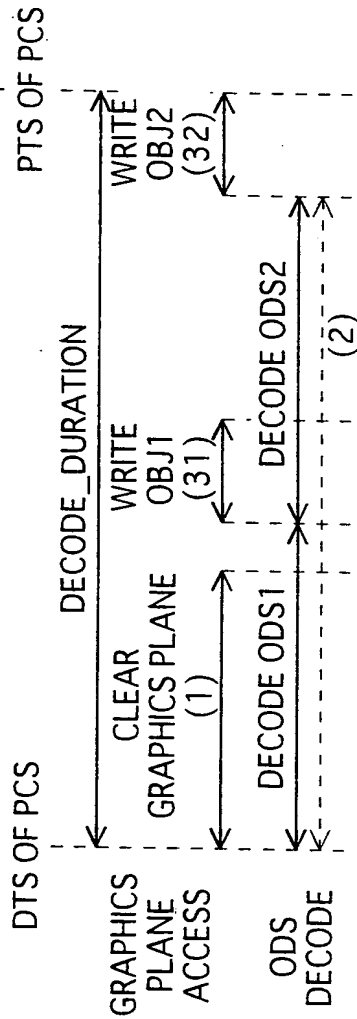


FIG.19C

DECODE_DURATION
 $= (1) + (31) + (32)$

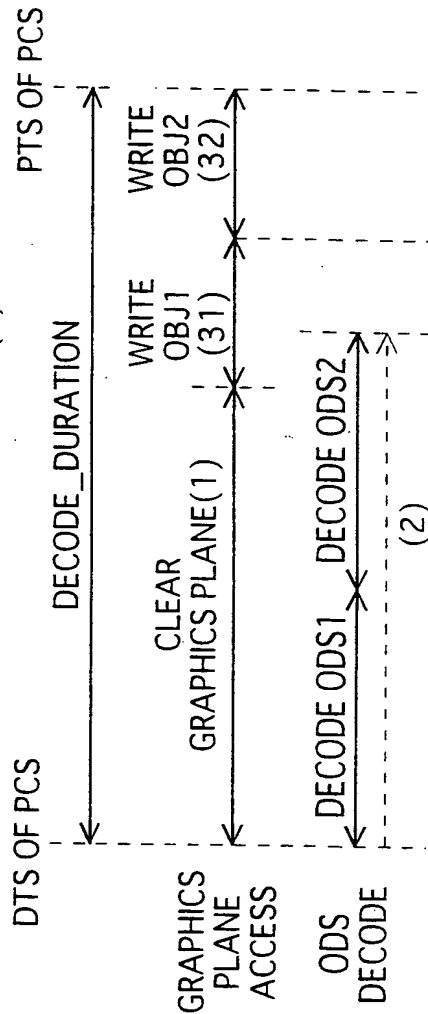


FIG.20

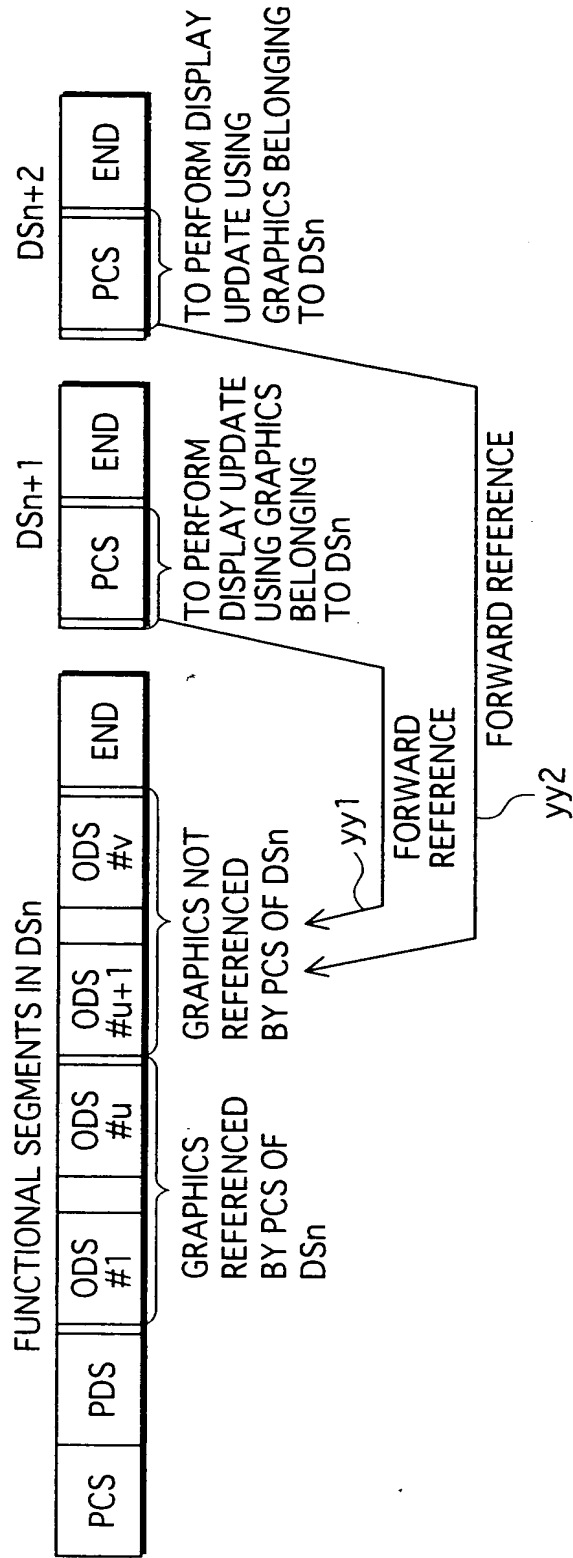


FIG.21A

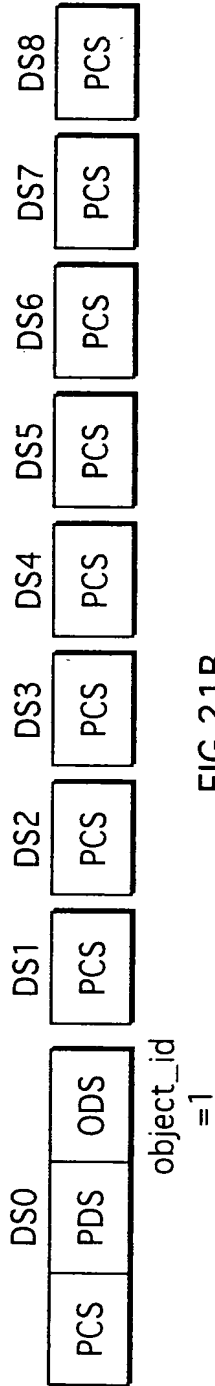


FIG.21B

	composition_object		PTS
PCS IN DS1	object_id_ref=1	object_horizontal_position=x1 object_vertical_position=y1	PTS=t1
PCS IN DS2	object_id_ref=1	object_horizontal_position=x2 object_vertical_position=y2	PTS=t2
PCS IN DS3	object_id_ref=1	object_horizontal_position=x3 object_vertical_position=y3	PTS=t3
PCS IN DS4	object_id_ref=1	object_horizontal_position=x4 object_vertical_position=y4	PTS=t4
PCS IN DS5	object_id_ref=1	object_horizontal_position=x5 object_vertical_position=y5	PTS=t5
PCS IN DS6	object_id_ref=1	object_horizontal_position=x6 object_vertical_position=y6	PTS=t6
PCS IN DS7	object_id_ref=1	object_horizontal_position=x7 object_vertical_position=y7	PTS=t7
PCS IN DS8	object_id_ref=1	object_horizontal_position=x8 object_vertical_position=y8	PTS=t8

FIG.22A

My heart is fluttering
 Object_id=1

FIG.22B

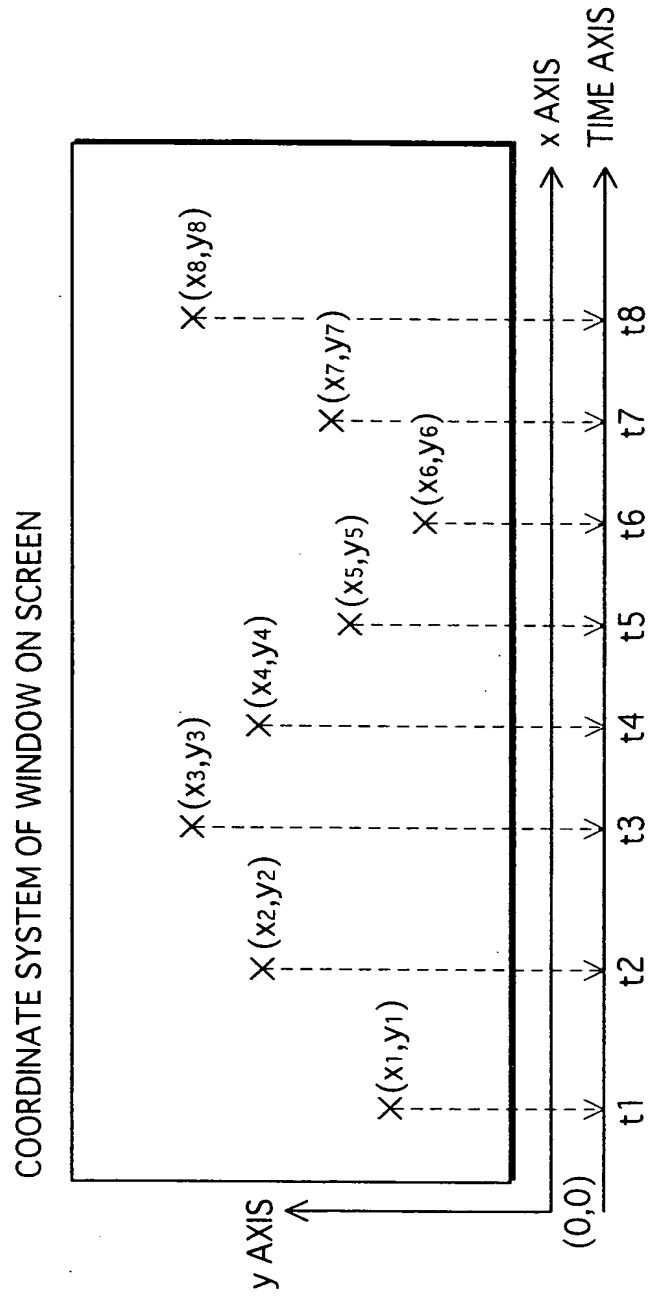


FIG.23

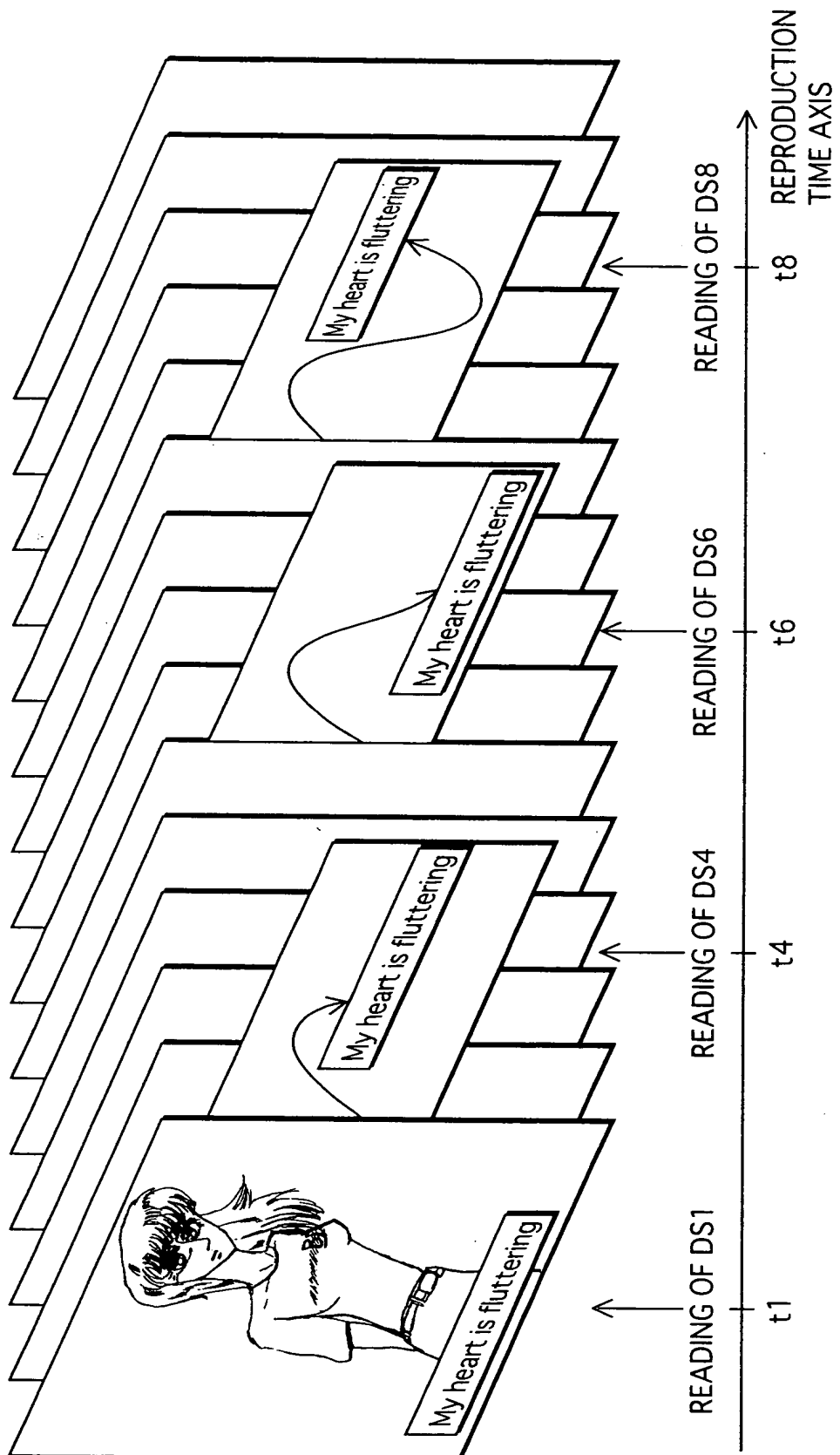


FIG.24

MOVEMENT OF GRAPHICS IN WINDOW

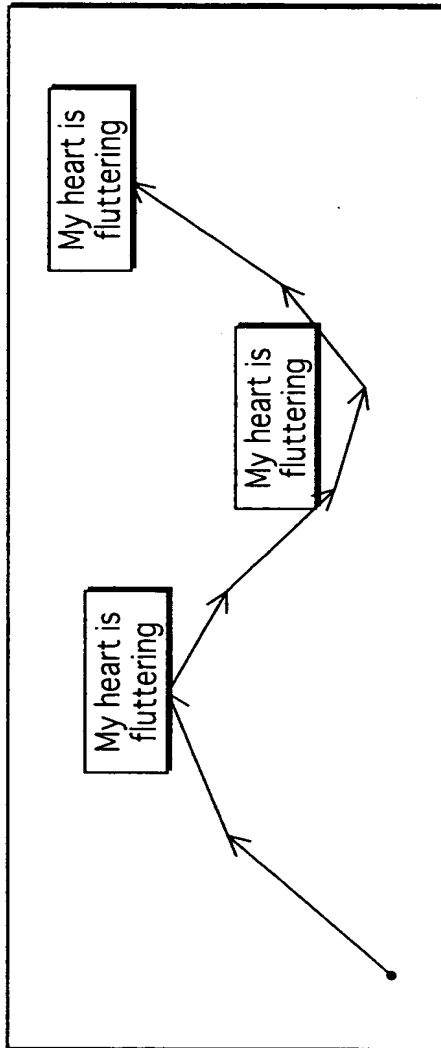


FIG.25

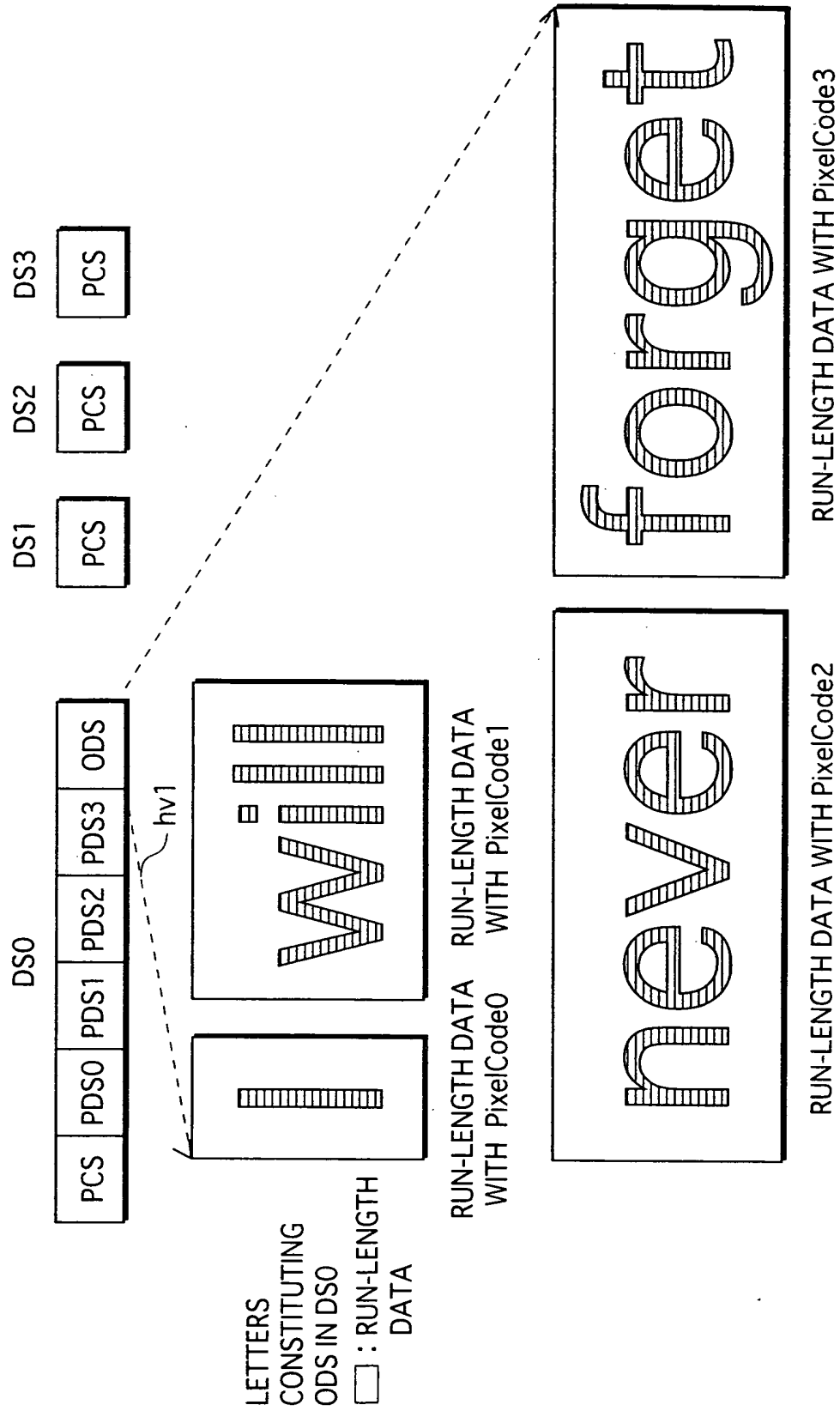


FIG.26A

FIRST LEVEL: PDS0 palette_id=0	PixelCode0 =RED	PixelCode1 =WHITE	PixelCode2 =WHITE	PixelCode3 =WHITE
SECOND LEVEL: PDS1 palette_id=1	PixelCode0 =RED	PixelCode1 =RED	PixelCode2 =WHITE	PixelCode3 =WHITE
THIRD LEVEL: PDS2 palette_id=2	PixelCode0 =RED	PixelCode1 =RED	PixelCode2 =RED	PixelCode3 =WHITE
FOURTH LEVEL: PDS3 palette_id=3	PixelCode0 =RED	PixelCode1 =RED	PixelCode2 =RED	PixelCode3 =RED

FIG.26B

FIRST LEVEL: PCS IN DS0	palette_update_flag=0	palette_id_ref=0	object_id_ref=1
SECOND LEVEL: PCS IN DS1	palette_update_flag=1	palette_id_ref=1	object_id_ref=1
THIRD LEVEL: PCS IN DS2	palette_update_flag=1	palette_id_ref=2	object_id_ref=1
FOURTH LEVEL: PCS IN DS3	palette_update_flag=1	palette_id_ref=3	object_id_ref=1

FIG.27

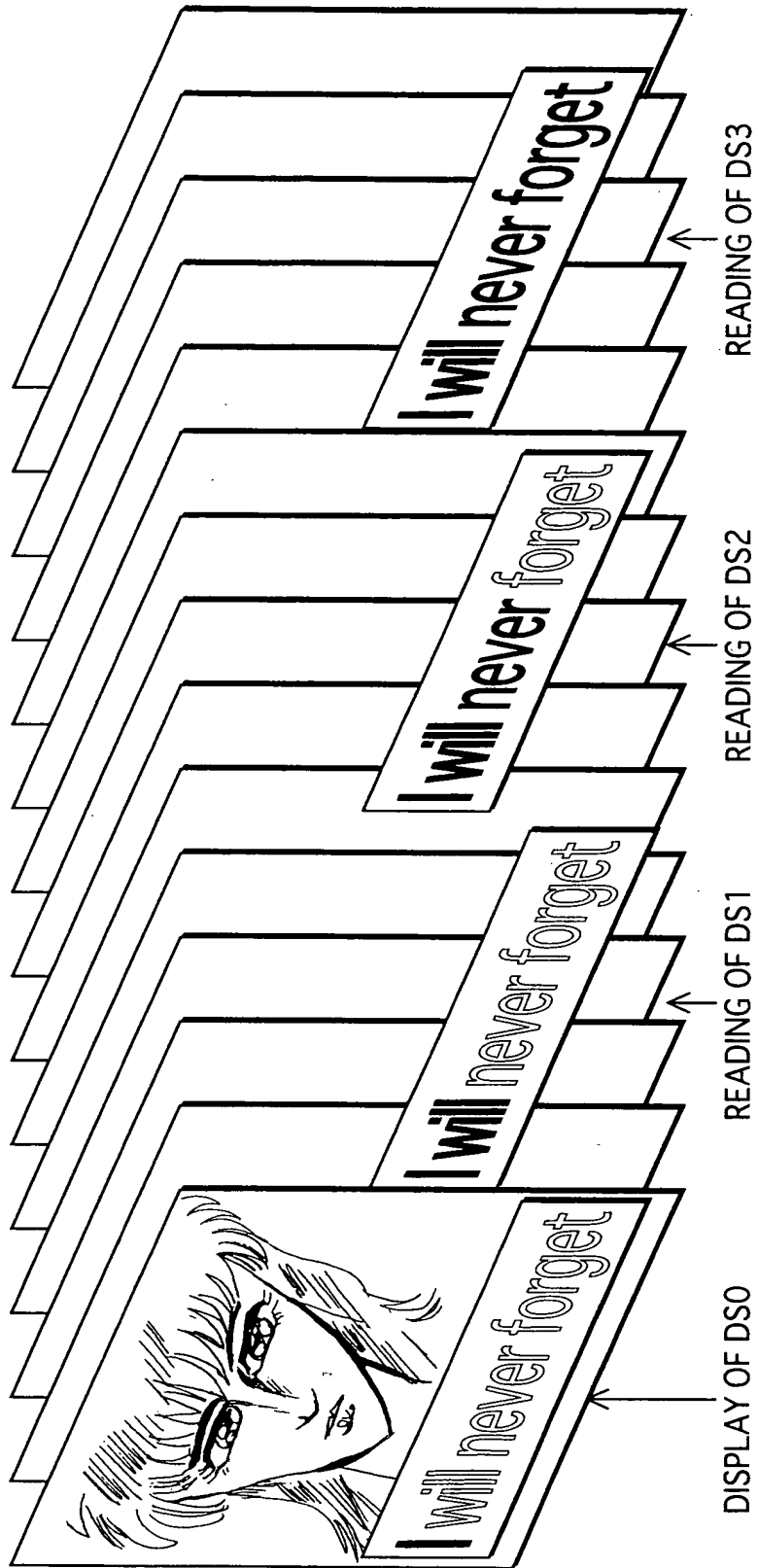


FIG.28

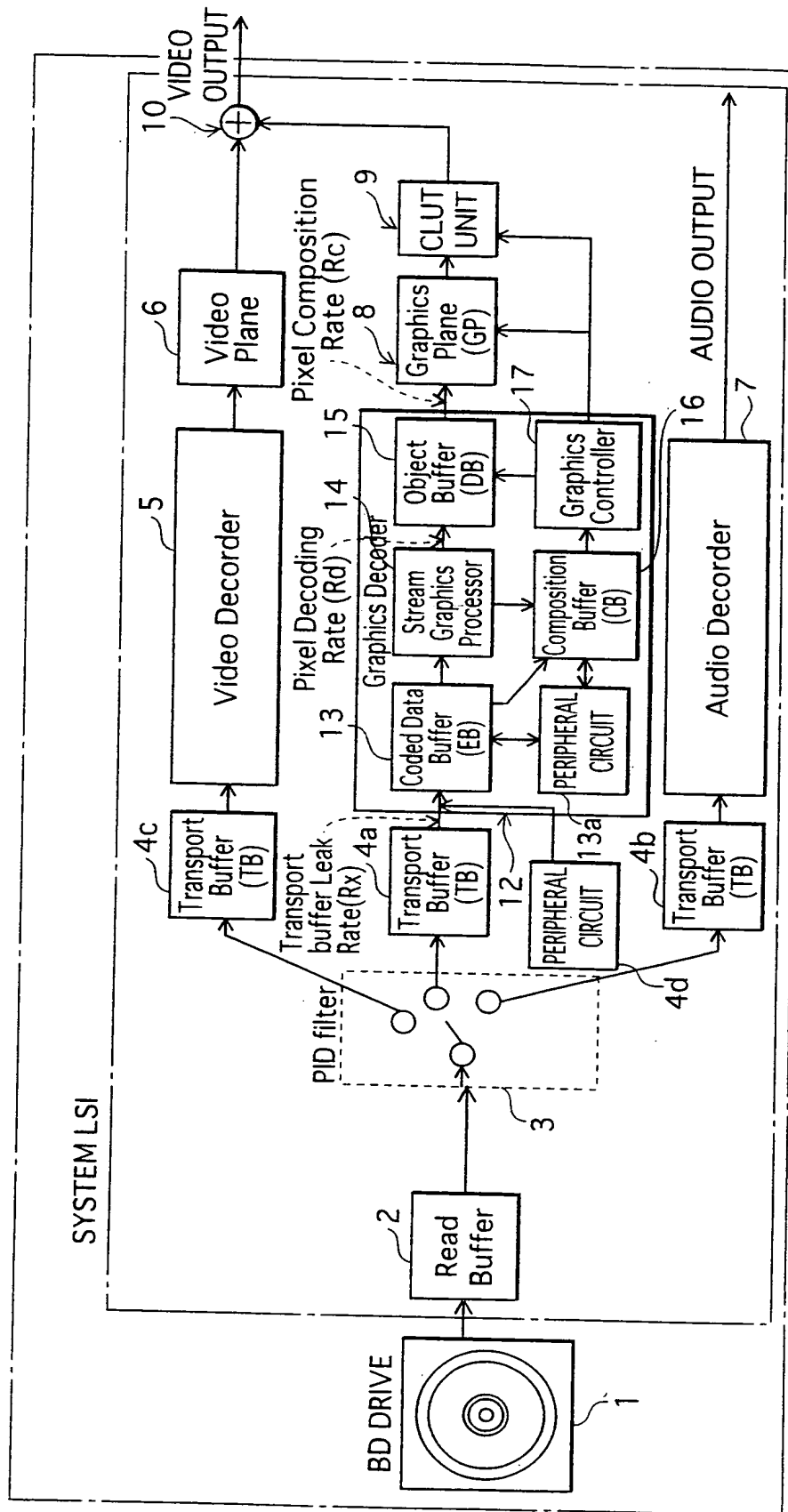
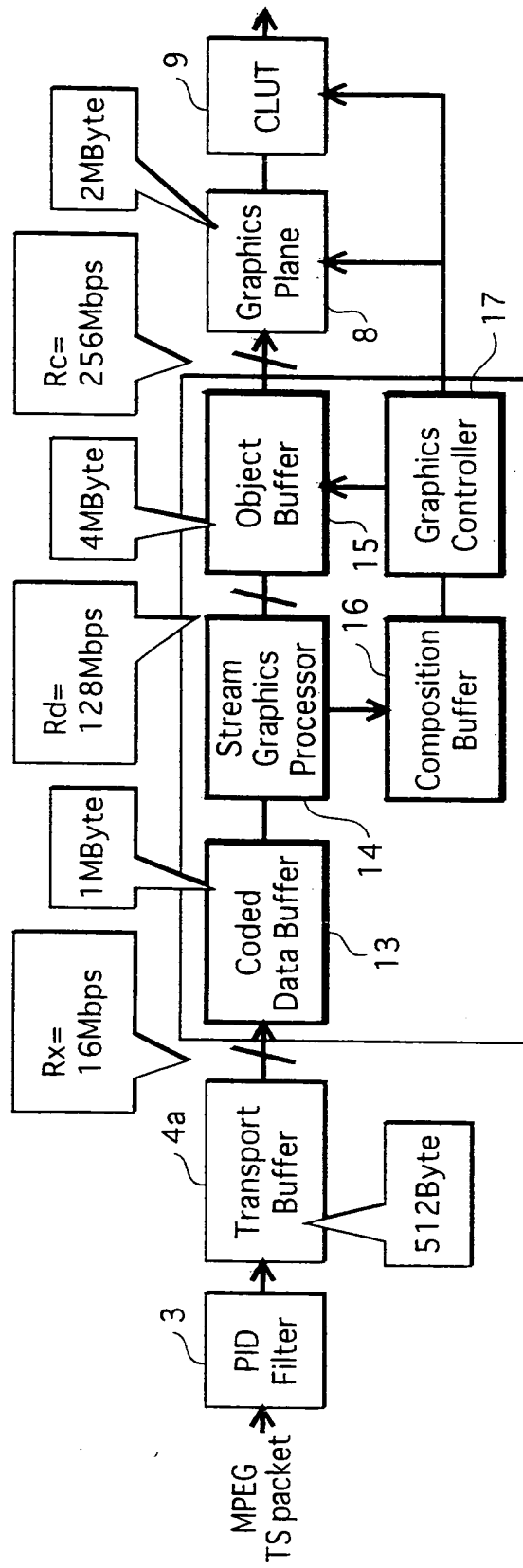
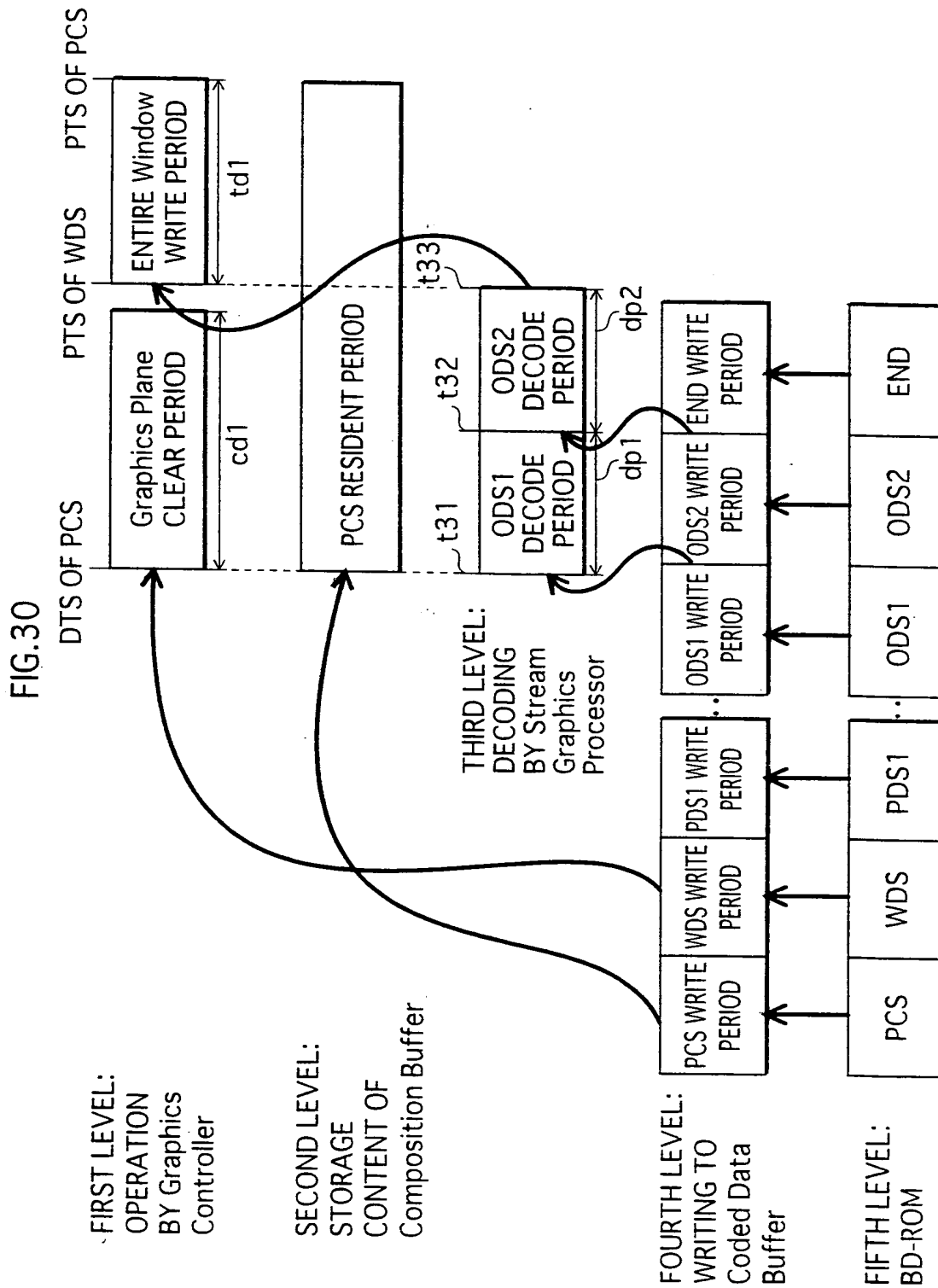


FIG. 29





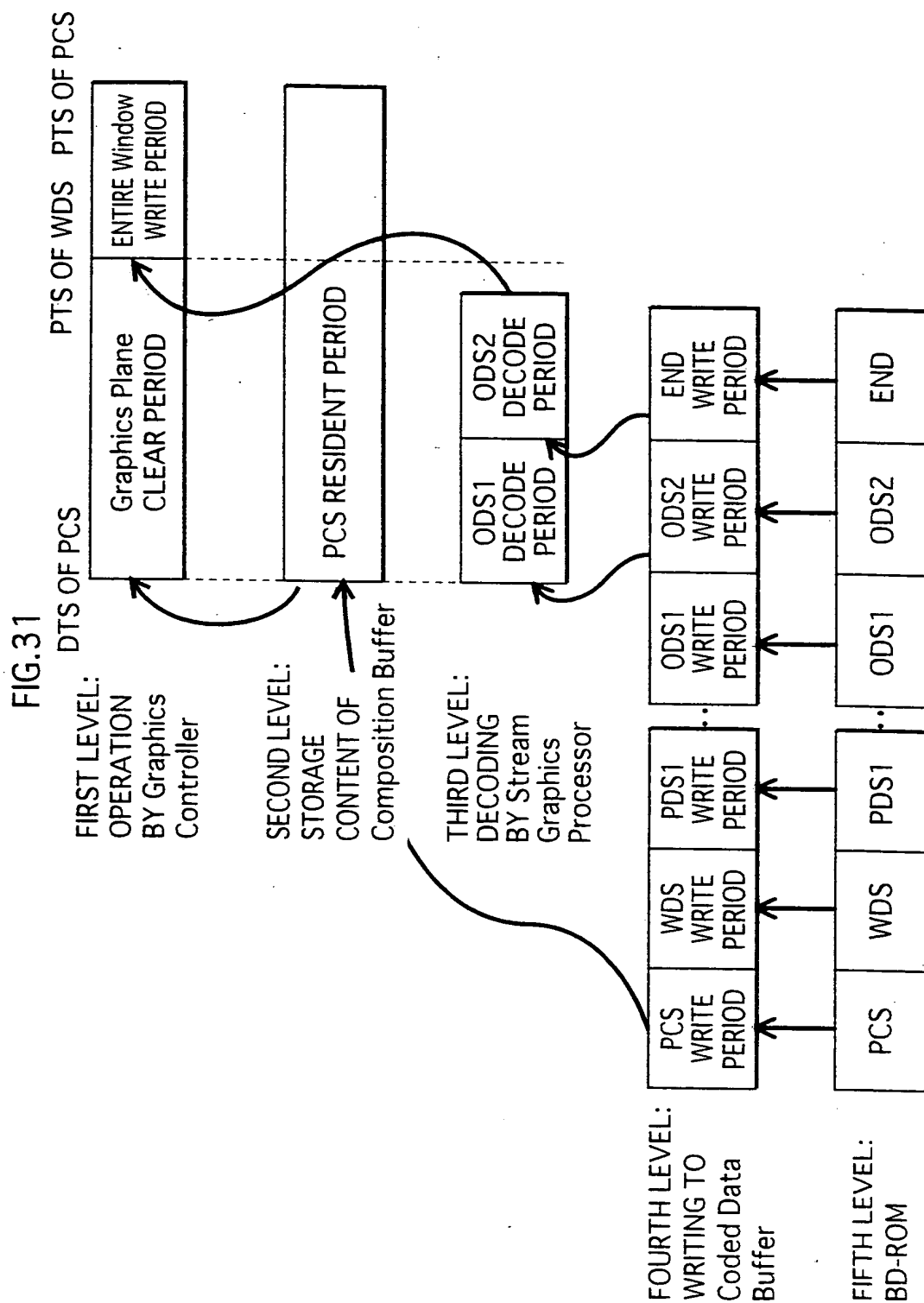


FIG. 32

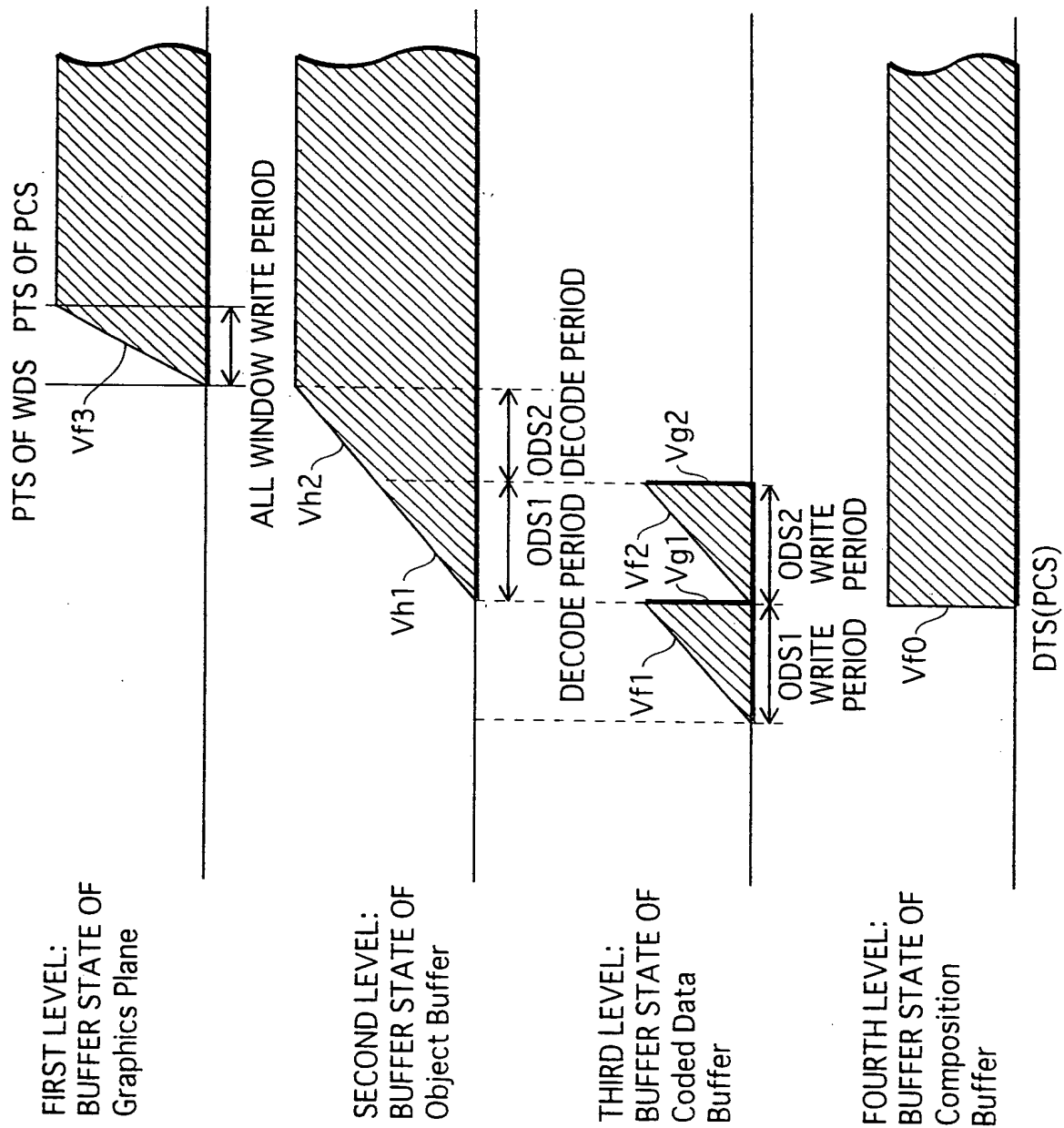


FIG.33

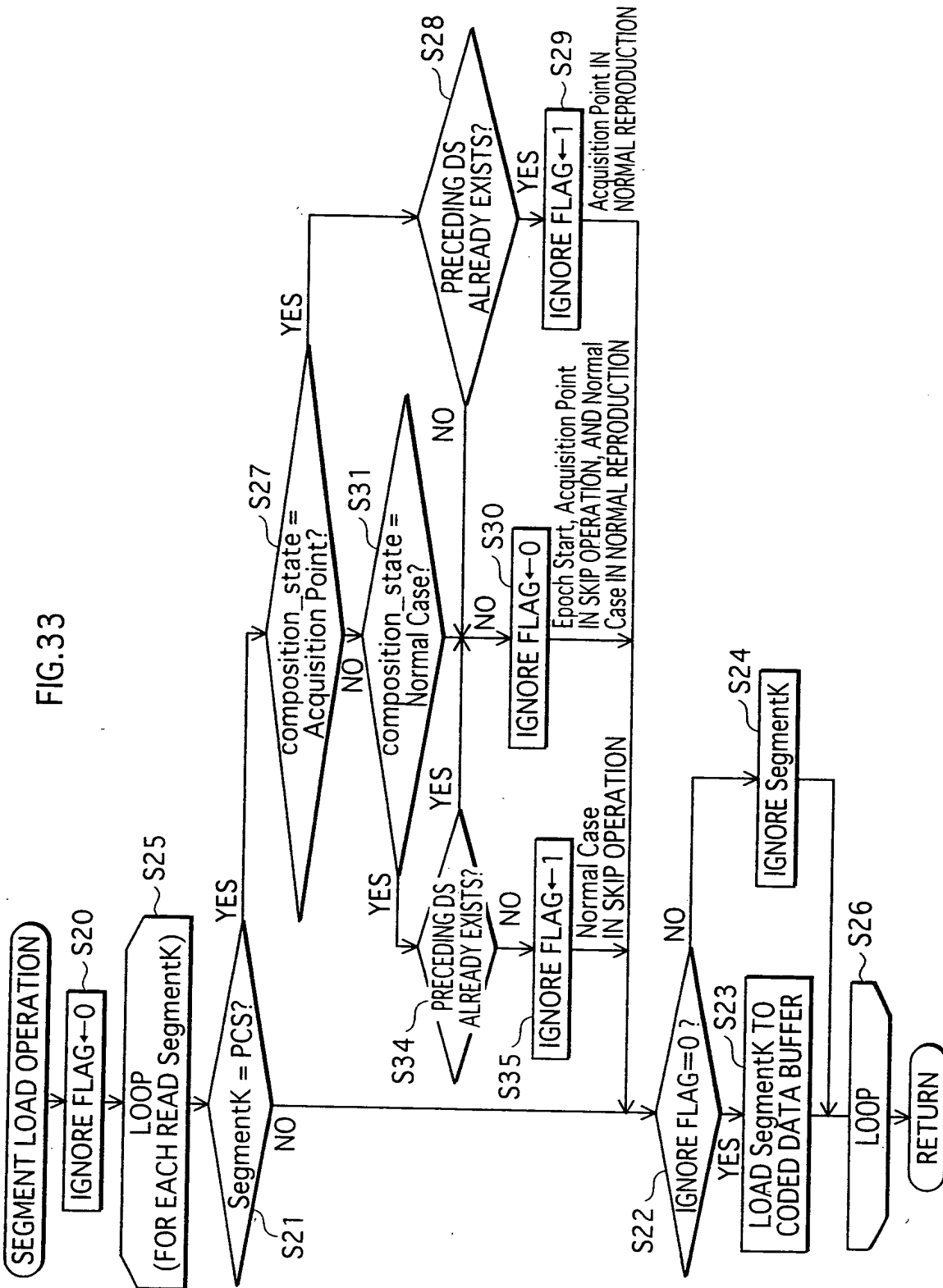


FIG.34

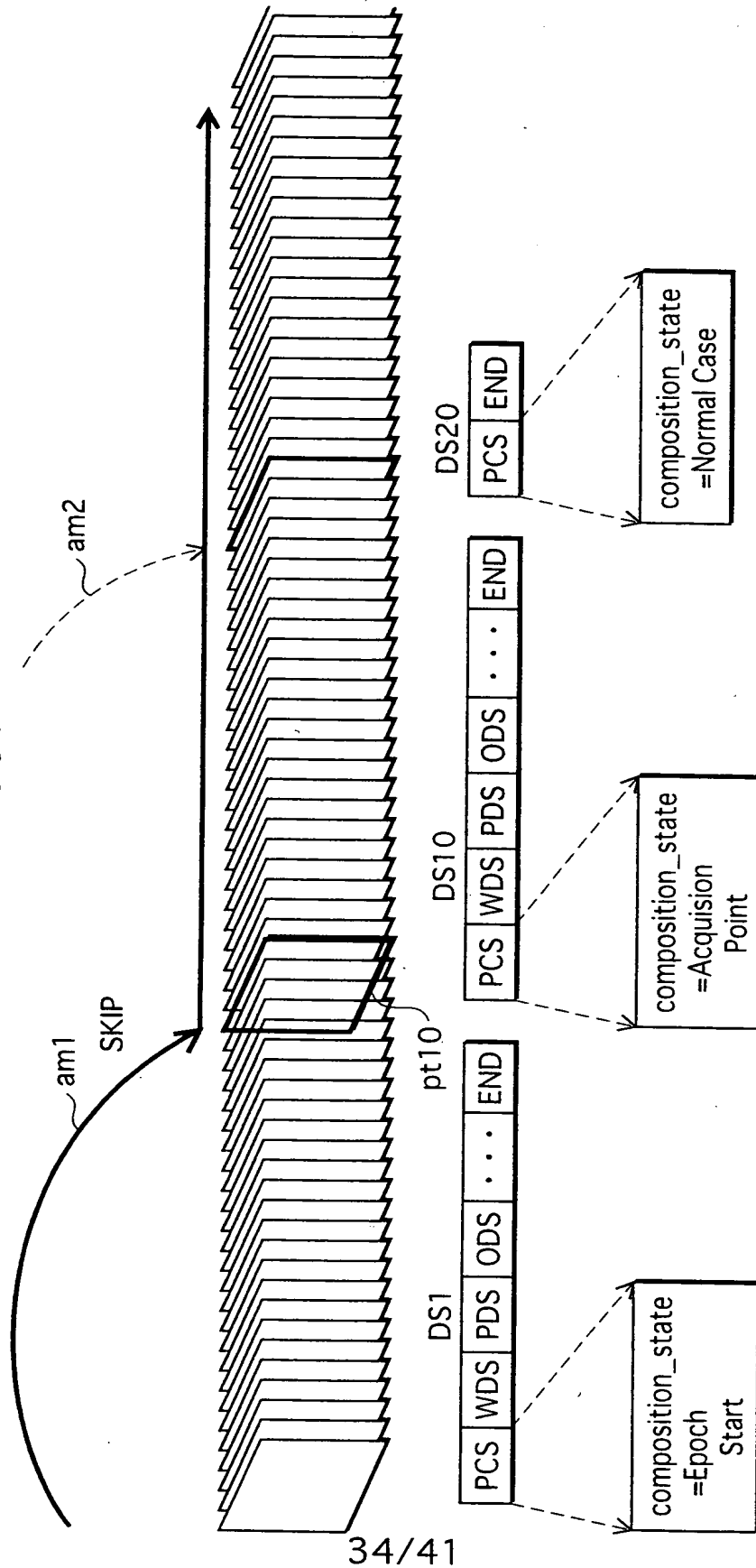


FIG.35

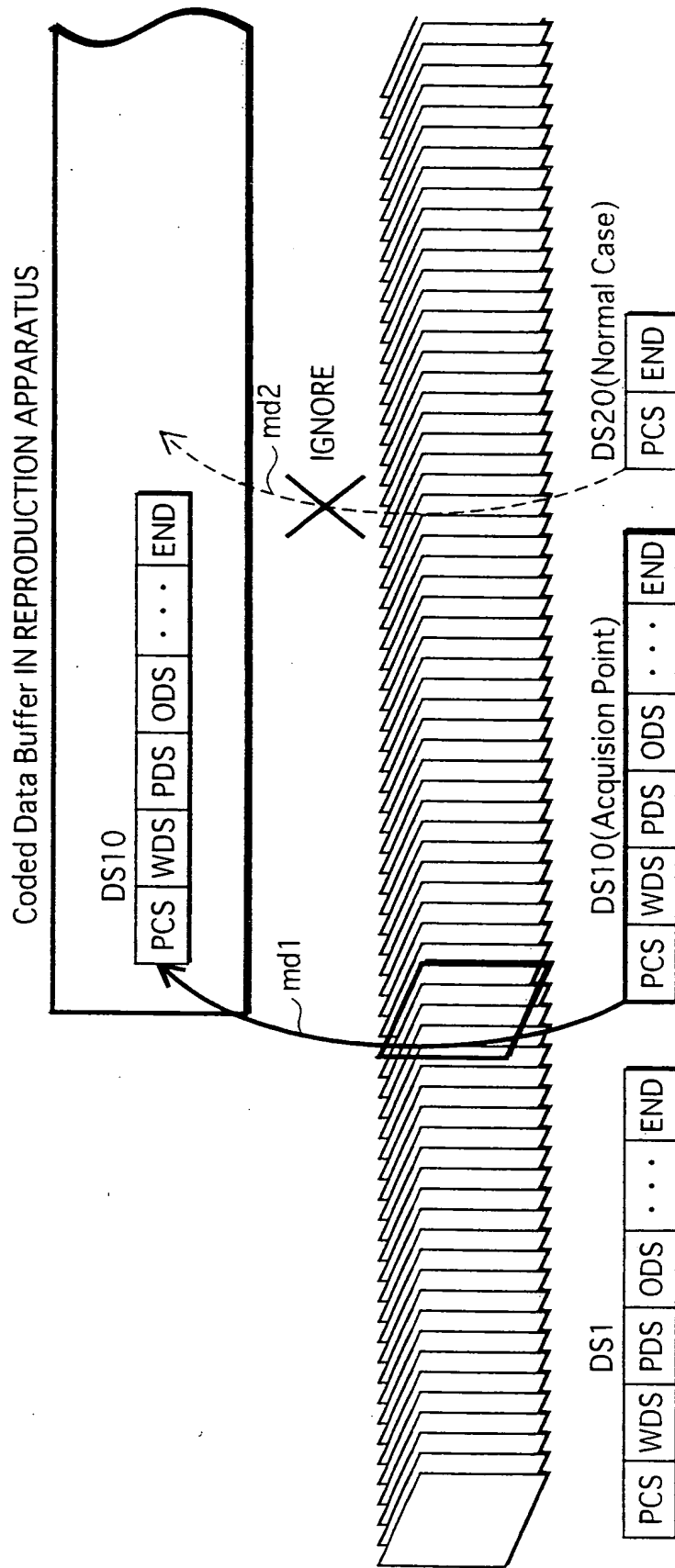


FIG.36

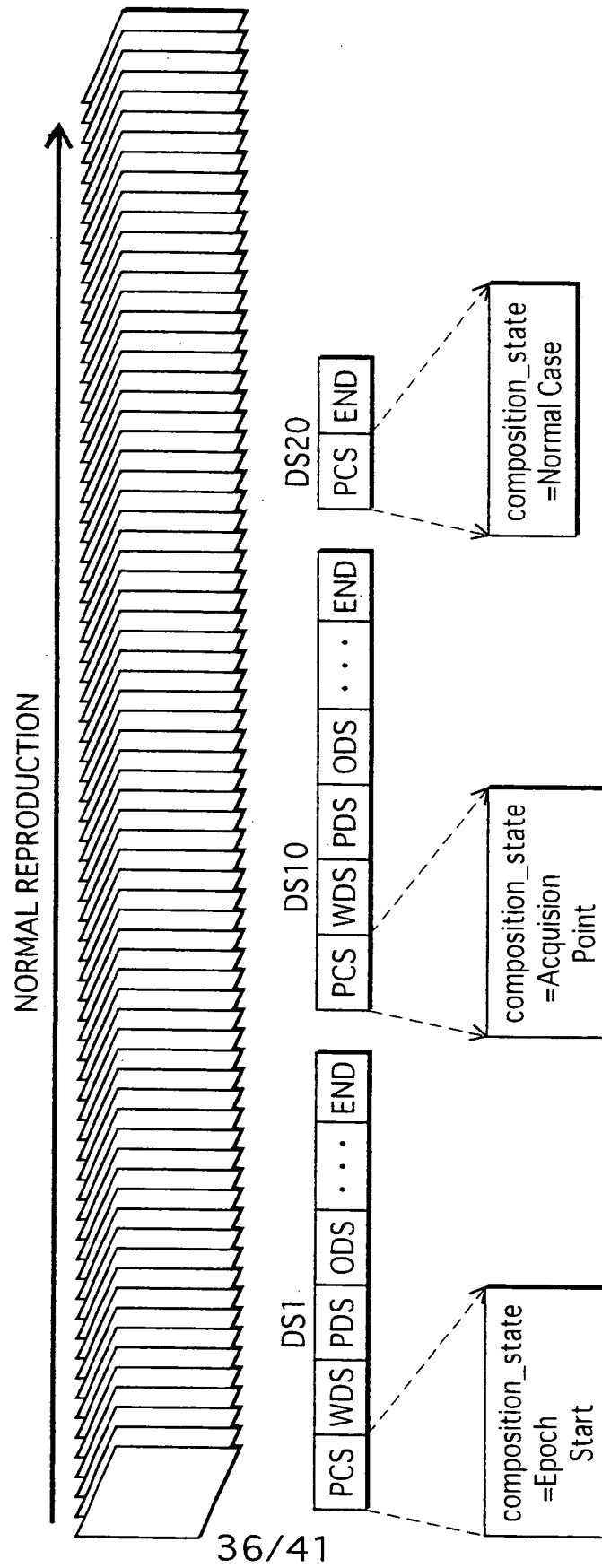
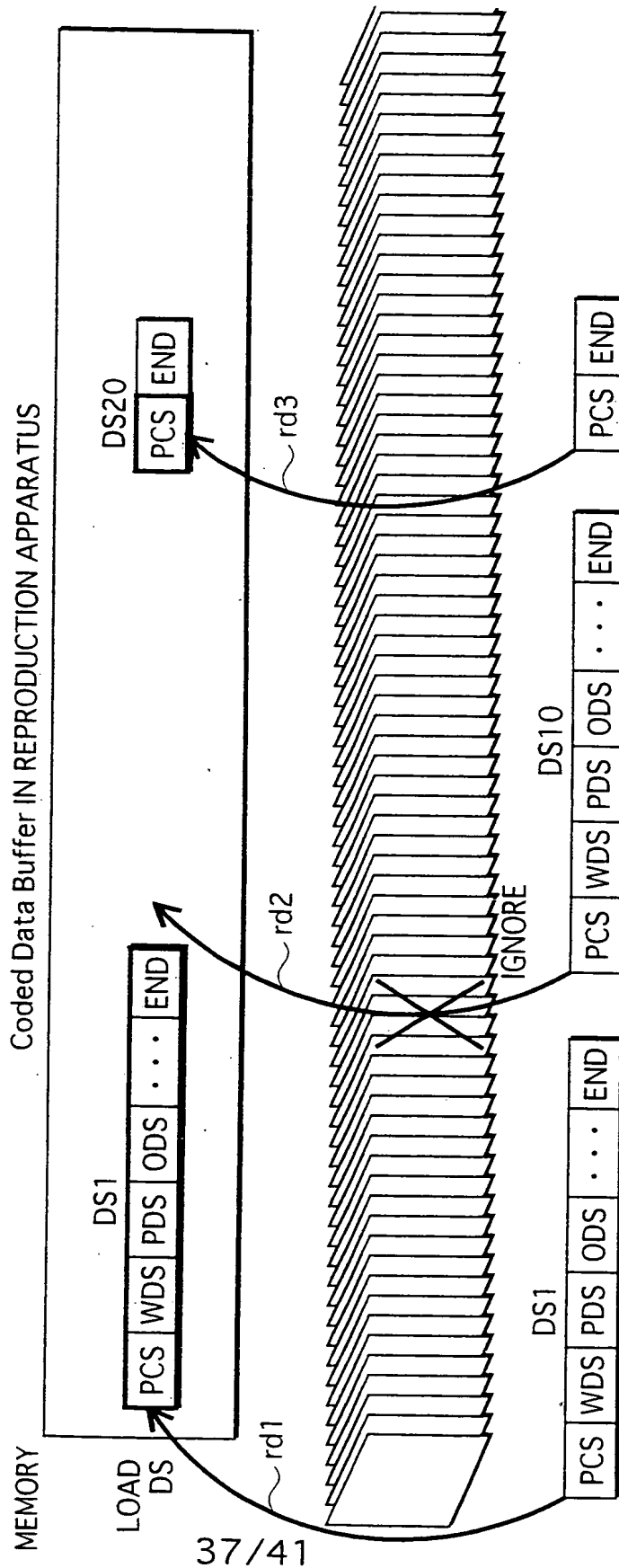


FIG.37



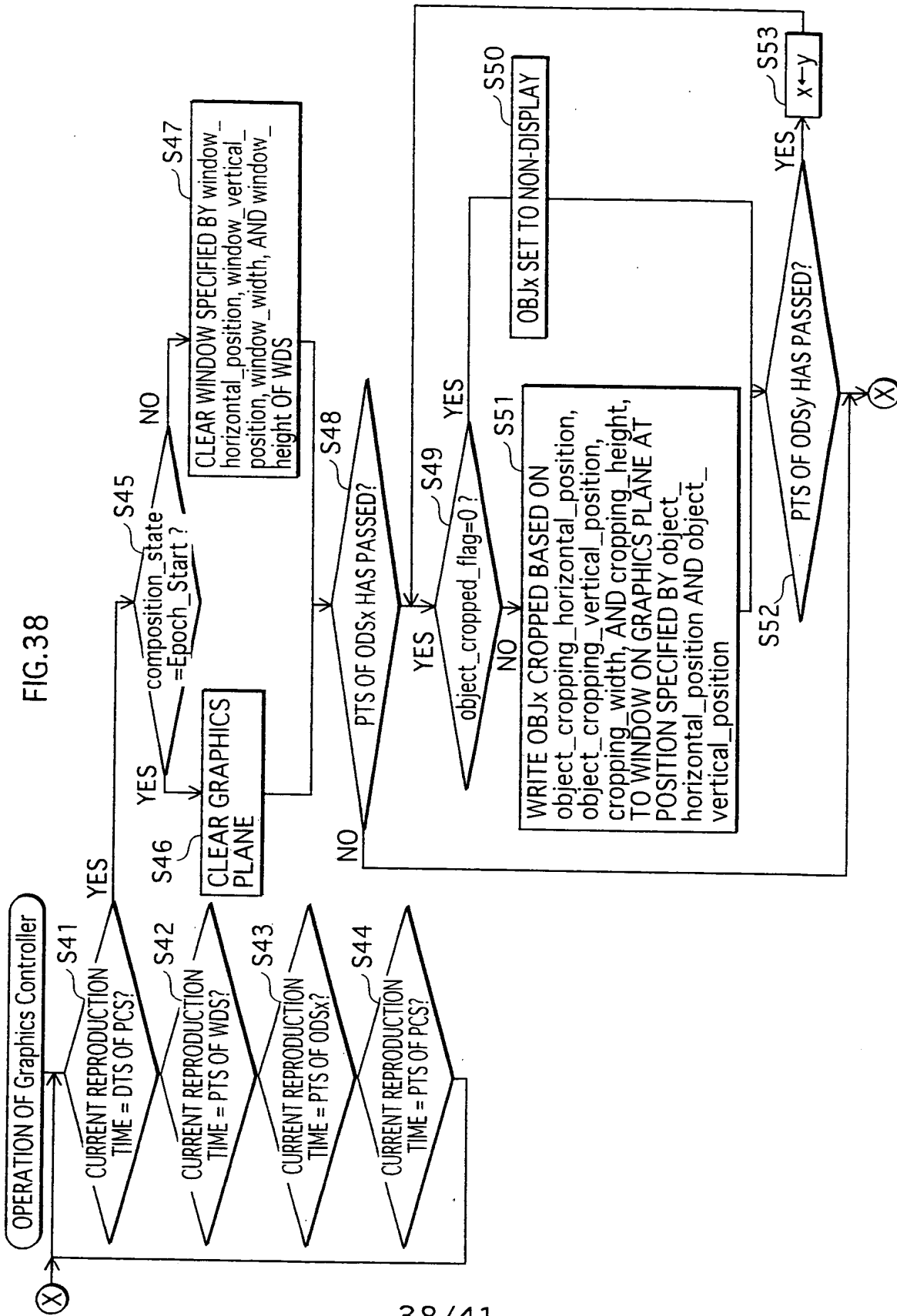


FIG. 39

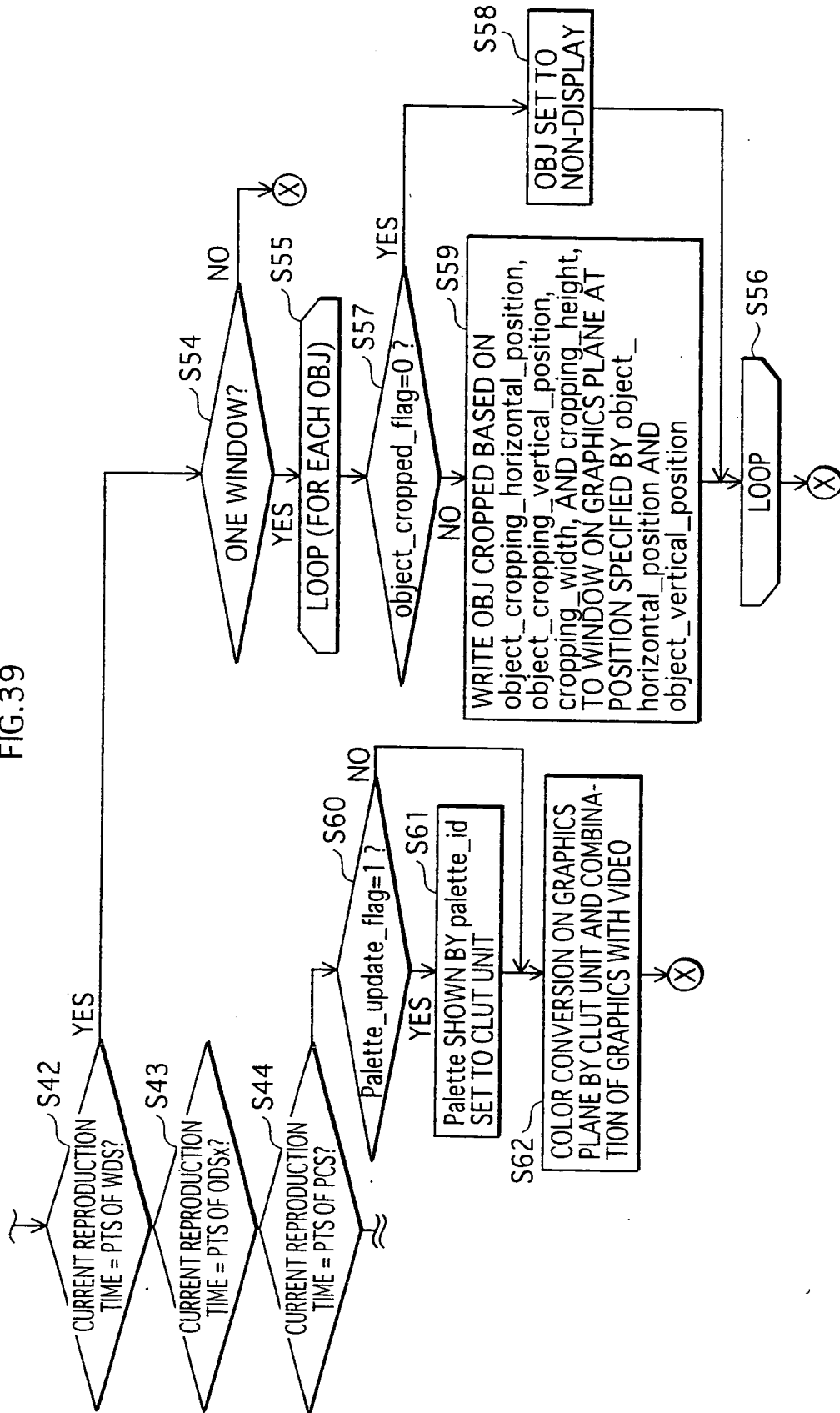


FIG. 40

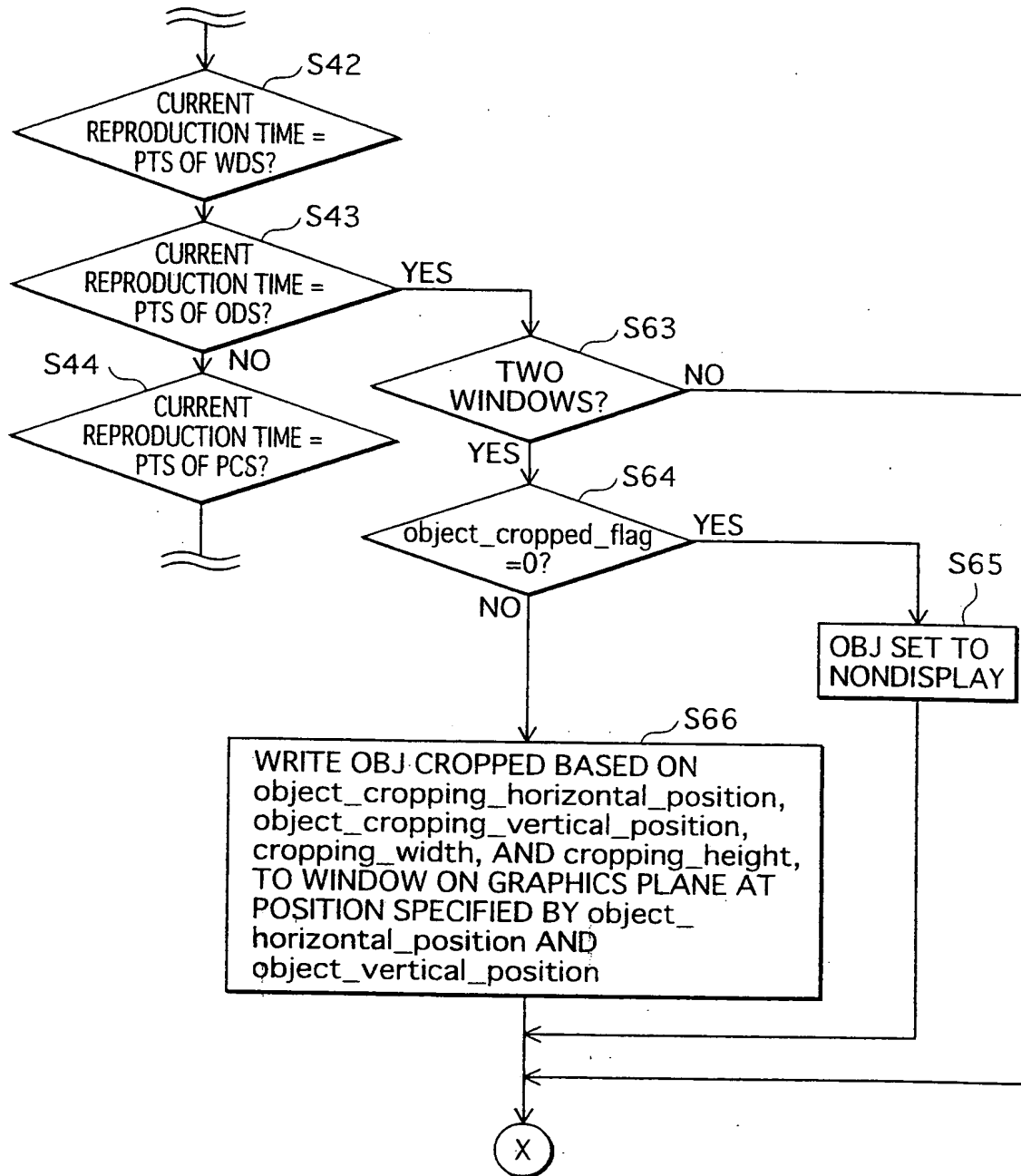


FIG. 41

